Theses and Dissertations	Thesis Collection

2011-03

Novel topic authorship attribution

Honaker, Randale J.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/5761

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

NOVEL TOPIC AUTHORSHIP ATTRIBUTION

by

Randale J. Honaker

March 2011

Thesis Co-Advisors: Craig Martell
Ralucca Gera

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*
25-03-2011

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
2010-01-01—2011-03-25

4. TITLE AND SUBTITLE
Novel Topic Authorship Attribution

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Randale J. Honaker

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. **IRB Protocol Number: n/a**

14. ABSTRACT

The practice of using statistical models in predicting authorship (so-called author-attribution models) is long established. Several recent authorship attribution studies have indicated that topic-specific cues impact author-attribution machine learning models. The arrival of new topics should be anticipated rather than ignored in an author attribution evaluation methodology; a model that relies heavily on topic cues will be problematic in deployment settings where novel topics are common. In order to effectively deal with novel topics, we create author and topic vectors and attempt to project out the topic influences from each document. Although our experiments did not validate our assumptions, they do point out a possible problem with a common assumption in authorship attribution research.

15. SUBJECT TERMS

Author attribution, novel topic, cross-validation, genre shift, vector projection, singular value decomposition, principal component analysis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	UU	97	

THIS PAGE INTENTIONALLY LEFT BLANK

NOVEL TOPIC AUTHORSHIP ATTRIBUTION

Randale J. Honaker
Lieutenant, United States Navy
B.S., Miami University, 2003

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN COMPUTER SCIENCE

AND

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
March 2011

Author:	Randale J. Honaker
Approved by:	Craig Martell Thesis Co-Advisor
	Ralucca Gera Thesis Co-Advisor
	Peter J. Denning Chairman, Department of Computer Science
	Carlos Borges Chairman, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The practice of using statistical models in predicting authorship (so-called author-attribution models) is long established. Several recent authorship attribution studies have indicated that topic-specific cues impact author-attribution machine learning models. The arrival of new topics should be anticipated rather than ignored in an author attribution evaluation methodology; a model that relies heavily on topic cues will be problematic in deployment settings where novel topics are common. In order to effectively deal with novel topics, we create author and topic vectors and attempt to project out the topic influences from each document. Although our experiments did not validate our assumptions, they do point out a possible problem with a common assumption in authorship attribution research.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Acknowledgements

First and foremost, I would like to thank my loving wife Amy without whose support I could never have pursued a dual degree.

To my advisors, thank you for pointing me in the right directions, helping me get excited about research and letting me run.

Thank you to the NPS NLP Lab for purchasing machines to allow our research to continue.

To my God who sustains me, I owe you my all.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

Modern authorship attribution began in ernest with the seminal study of Mosteller and Wallace [1] in 1964 where they investigated the "Federalist Papers." In the last several decades, the complexity of authorship attribution methods has increased significantly, moving from simple statistics on sentence length to treebanks showing long-distance dependencies. Authorship attribution researchers build machine learning classification models or rule-based systems identifying the author of an anonymous text given undisputed knowledge of various communications written by that particular author. The earliest (as well as continuing) efforts in the field looked at the authorship of historically interesting documents. Today, interest in the field is additionally motivated by fairness and public welfare concerns: plagiarism detection and identifying authors in a criminal investigation or intelligence setting.

The explosion of technology and availability of digital texts has greatly simplified research in one aspect and revealed some weaknesses as well. When authorship attribution methods were performed by hand, it was very time consuming to apply a method across several corpora. With the plethora of digital documents available today this is not an issue, and researchers are able to relatively easily apply a model that proved successful in one area to another area. While the techniques have become more and more sophisticated, the area of research has mostly remained limited to areas where all candidate authors are writing on the same topic. Having a single topic in a corpus then, may be an oversimplification that can produce misleading results. We propose that an author model should be able to accurately model the unique style of the author. If a model can capture an author's unique style, then the same author model should be able to be applied across multiple topics and accurately classify the author. We posit that much of the success that past researchers have had in performing authorship attribution is due to their holding the topic constant and not to the ability of their model to accurately capture the subtle uniqueness of a particular author.

1.1 Research Applications

It is increasingly important to determine who actually wrote texts written by unknown authors. Authorship attribution is particularly useful for law enforcement and the military as they attempt to determine whether a person suspected of a crime is actually the author of some incriminating document. It is easy to imagine a situation where a person, call him John, is suspected of being a terrorist. Suppose we have access to a sufficient number of samples of John's writing such as some of his work emails, blog posts from a hobbyist site, and maybe some of his home emails. Let us also suppose that we suspect John of being the author of some anonymous posts on a known jihadist website. What we want to know is if we can tie John to the anonymous posts using information from his other writing.

Our goal is to modify existing techniques in order to allow models built of an author to be applied across topic domains. We want a method to be able to say, with some degree of certainty, that John either is or is not the author of the anonymous blog posts.

1.2 Research Question

Our question was, "Can we construct feature-count vectors of documents and topics in such a way that we can project out the topic influence on the document and create an author vector that is topic independent?"

We attempted to answer this question by applying known methods from linear algebra in a new way, projecting document vectors onto topic vectors. A subset of the New York Times (NYT) Annotated Corpus was used for our hypothesis testing since each article had a single author and was written on a single topic. We converted the documents into vectors, created topic vectors, and attempted to project out the topic influence on each of the documents. What we hoped to have constructed were vectors that represented the author and were not influenced by the topic.

Some assumptions made were: each author has a unique way of writing that does not vary across topics, the author's style and the topic are completely independent, author and topic vectors are appropriately represented as feature-count vectors.

1.3 Results

We were unable to construct feature-count vectors of documents and topics in such a way that we could project out the topic influence on the document and create an author vector that is topic independent. The method of projecting unigram document vectors onto topic vectors reduced the weighted accuracy from 80% to 54% when compared against a standard bag-of-words model. While our methods did not produce author models that can be successfully applied across topic domains, we did show that some of the underlying assumptions often made in the field of natural language processing (NLP) may not be valid assumptions.

1.4 Organization of Thesis

In order to investigate the research question, this thesis is organized as follows:

- Chapter 1 discusses the motivation and provides an overview of the methods used in this research.
- Chapter 2 discusses prior and related work in the fields of linear algebra and authorship attribution.
- Chapter 3 contains a description of the methods used to prepare the data and conduct experiments.
- Chapter 4 contains the results of the experiments and analysis.
- Chapter 5 contains the summary and possible areas for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Prior and Related Work

2.1 Prior Work

The field of authorship attribution is concerned with trying to discover who, from a group of candidates, is the author of some text. One of the most famous examples of authorship attribution was conducted by Mosteller and Wallace [1] where they applied their statistical analysis to the *Federalist* papers. Methods of authorship attribution have evolved from relatively simple techniques such as statistical analysis on length of sentences [2] to applying complicated machine learning algorithms to stylometric features of a text [3]. More recently, Gehrke [4], used Bayesian methods to perform authorship attribution in blogs. Gehrke concluded that topic cues overwhelm classifiers and make cross topic authorship attribution difficult. For a more thorough review of authorship attribution techniques see Stamatatos *et al.* [5].

2.2 Related Work

Performing authorship attribution under novel topic scenarios has largely been ignored by the NLP community due to its difficulty. There have been a few studies conducted where novel topics were considered, however, we believe the number of topics they considered was too small.

2.2.1 Topic Influence Studies

Mikros and Argiri [6] attempted to test whether or not stylometric features used in authorship attribution are topic neutral. They used the following stylometric features:

- Lexical richness variables
- Function words (10 most frequent modern Greek words)
- Sentence level measures
- Word level measures
- Character level measures

Their corpus consisted of 200 modern Greek newswire articles that were completely balanced across two authors and two topics. They concluded that there is a significant correlation between the stylometric features, and the topic text using stylometric features to perform authorship attribution should be done with caution.

A second study conducted by Koppel, Schler, and Bonchek-Dokow [7] looked at the depth of difference between two example sets across topics. They used a corpus of 21 books written by ten authors. They noticed that, using traditional authorship attribution methods, it was difficult to accurately determine the author of a text when the topic of the training documents differed from the topic of the testing documents. To solve this problem, they invented a technique they call unmasking. The intuition behind unmasking is to iteratively remove the features that are most useful for distinguishing between X and Y and to gauge the speed with which cross-validation accuracy degrades as more features are removed. Their hypothesis was authors use a consistent style across topics and when the most distinguishing features are removed from document X, it is more difficult to distinguish it from document Y, thus, indicating that they were both written by the same author. They concluded that it is more difficult to distinguish writings by the same author on different topics than writings by different authors on the same topic.

A third study conducted by Corney [8] concluded that topic has no effect on authorship attribution and stylometric features are sufficient for classifying email messages. He used a corpus of 156 e-mail messages from three authors writing about three topics. He picked one of the topics to create a model of each author and tested the model on the remaining two topics using support vector machines. He identified an author 85% of the time across topics. He noted that much of his success came from the length and structure of e-mail messages. Since the most topic specific cues in an email are in the subject line, by looking only at the body of the e-mail the impact of topic-specific words may be negligible.

In contrast to results obtained by Corney, the fourth study, by Madigan *et al.* [9], tested the effect of topic on authorship attribution with 59 Usenet postings by two distinct authors on three distinct topics. Just as with Corney, they constructed a model of each author on one of the three topics and tested for authorship on postings written about the remaining two topics. They obtained poor performance using a unigram model, but their bi-gram parts-of-speech (pairs of consecutive parts of speech) model proved to be one of the best among the tested possibilities.

Finally, the fifth study conducted by Baayen *et al.* [10], used principal component analysis (PCA) and linear discriminant analysis (LDA) to evaluate the effectiveness of grouping text by author, using stylometric features. Their data set consisted of 576 documents written by eight students. Each student wrote a total of 24 documents in three different genres about three different topics. They found that compensating for the topic imbalance coverage led to increased performance in a cross-validation.

2.3 Separating Author from Topic

What we see from the prior work leads us to conclude that traditional authorship attribution techniques are picking up on topic cues in the text and those cues are overwhelming the authorship signal when we move across topic domains [11]. Our hypothesis is that each author has a style that defines him which does not vary across the topics he is writing about. We model this as a vector-space problem. To begin with, we break all documents up into classes based on their topic. If we think of each document i in class j as a vector $\vec{D_{i,j}}$ and assume we can construct a topic vector $\vec{T_j}$ for each class j, then what we are looking for is some residual vector $\vec{R_j}$ which is "as close to" $\vec{D_{i,j}}$ as possible but also has "nothing in common" with (orthogonal to) $\vec{T_j}$. Formally stated, we hypothesize that the solution to Equation (2.1), subject to $\vec{R_i} \cdot \vec{T_{i,j}} = 0$, will be a vector that represents the author's unique style. We removed the subscripts in the equation for clarity.

$$\arg\min_{\vec{R}} ||\vec{D} - \vec{R}||^2 \tag{2.1}$$

We now prove that the solution to this problem is found using standard vector projection [12]. Consider a document vector $\vec{D}$ and a topic vector $\vec{T}$. We desire a residual vector $\vec{R}$ that is "as close to" $\vec{D}$ as possible but also orthogonal to $\vec{T}$. "As close to" will be formalized as the squared error objective function $|| \cdot ||^2$.

That is, we desire to minimize Equation (2.1) subject to

$$\vec{R} \cdot \vec{T} = 0.$$

We hypothesize the existence of a vector $\vec{P}$ that subtracted from $\vec{D}$ gives us the solution vector

$\vec{R}$. Since $\vec{R} = \vec{D} - \vec{P}$, the problem may be reformulated:

$$\arg\min_{\vec{P}} ||\vec{P}||^2$$

subject to

$$(\vec{D} - \vec{P}) \cdot \vec{T} = 0.$$

To solve this problem, we will use the method of Lagrange multipliers [13], defining the function $f(\vec{P}, \lambda)$ as follows.

$$f(\vec{P}, \lambda) = ||\vec{P}||^2 - \lambda((\vec{D} - \vec{P}) \cdot \vec{T})$$

Taking the derivative of f with respect to $\vec{P}$ and λ gives the following set of equations which we set equal to zero.

$$0 = 2\vec{P} + \lambda\vec{T} \tag{2.2}$$

$$0 = -\vec{D} \cdot \vec{T} + \vec{P} \cdot \vec{T} \tag{2.3}$$

Solving Equation (2.2) for $\vec{P}$ we get:

$$\vec{P} = \frac{-\lambda}{2}\vec{T}$$

We substitute this value for $\vec{P}$ into Equation (2.3) and solve for λ.

$$0 = -\vec{D} \cdot \vec{T} + \frac{-\lambda}{2}\vec{T} \cdot \vec{T}$$

$$\lambda = \frac{-2\vec{D} \cdot \vec{T}}{\vec{T} \cdot \vec{T}}$$

With λ solved, we can back substitute into Equation (2.2) revealing a closed form for $\vec{P}$:

$$0 = 2\vec{P} + \frac{-2\vec{D} \cdot \vec{T}}{\vec{T} \cdot \vec{T}}\vec{T}$$

$$\vec{P} \;=\; \frac{\vec{D} \cdot \vec{T}}{\vec{T} \cdot \vec{T}} \vec{T} \tag{2.4}$$

Equation (2.4) is equivalent to Equation (2.7) and thus our hypothesized $\vec{P}$ is simply vector projection found in linear algebra textbooks.

2.3.1 Novel Topic Cross-Validation

We wish to develop an author model that is independent of the topic. In order to test our model, we need a situation where we have a new document written by a known author on a never before seen topic. Novel topic cross-validation simulates a scenario where we try to perform author attribution when novel topics appear. This simulation is accomplished by performing a leave-one-topic-out n-fold cross validation where n represents the total number of topics in the data set. In each of the n folds, we test on all documents pertaining to one topic and train on all other documents pertaining to the remaining $n - 1$ topics.

2.4 Features

In the field of natural language processing (NLP), the method of transforming a piece of natural language text into a vector appropriate for machine learning algorithms is determined by what types of features you would like to use. There are various features available to choose from that are well defined and understood in the NLP community. Here we outline the feature sets used in our experiments.

2.4.1 Unigrams and Bigrams

The unigram model for representing text is commonly referred to as a “bag of words” model. Word order information is lost when using this model. A document is represented as a collection of (word, count) tuples where each ‘word’ is a word from the document and ‘count’ is the number of times that word appears in the document. There are as many tuples in the representation as there are unique words in the text. Bigrams are created in a similar manner except that adjacent word pairs replace words in the tuples [14]. These unigrams and bigrams can also similarly be created using characters rather than words.

2.4.2 Gappy Word Bigrams

In addition to traditional adjacent word bigrams, we can define various other types of word bigrams, for example gappy bigrams or orthogonal sparse bigrams. Gappy bigrams are defined by creating a bigram from all words that have a distance between them that is less than or equal to some pre-defined maximum distance [15]. The distance between two words, a and b, is equal to the number of other words in between word a and word b. Traditional bigrams are a subset of gappy bigrams where the maximum distance between words is zero. As an example, the gappy bigrams produced by the phrase “the big purple dog” with a gap of two is shown in Figure 2.1. Gappy bigrams often include tags to mark the beginning and end of a phrase or sentence.

{the_big, the_purple, the_dog,
big_purple, big_dog, purple_dog}

Figure 2.1: Gappy bigrams formed from the phrase “the big purple dog”.

These typical tags were excluded in our constructions of gappy bigrams since one of our preprocessing steps is to remove capitalization and punctuation from the documents in our corpus. Gappy bigrams are a variant of string kernels except that string kernels [16] apply to characters rather than to words. They are also very similar to orthogonal sparse bigrams.

2.4.3 Orthogonal Sparse Bigrams

Orthogonal sparse bigrams (OSB) were initially defined by Siefkes *et al.* [17] in a similar manner to gappy bigrams. The distinction between gappy bigrams and OSBs is, with the initial definition of OSBs, the maximum distance is fixed at five. Defined in this manner, OSBs are a proper subset of sparse binary polynomial hashing (SBPH). The definition was later refined by Cormack *et al.* [18] to explicitly include the distance between two words and allow the maximum distance to be a fixed number, usually less than six. The distance between words is defined the same way as in gappy bigrams. OSBs produced by the phrase “the big purple dog” with a maximum distance between words of two is shown in Figure 2.2.

{the(0)big, the(1)purple, the(2)dog,
big(0)purple, big(1)dog, purple(0)dog}

Figure 2.2: OSBs formed from the phrase “the big purple dog”.

2.5 Entropy

We refer to entropy in the information theoretic sense Shannon [19] introduced, as a measure of uncertainty present in a given distribution. The higher the entropy the greater the uncertainty and vice versa. The entropy, H, of some distribution P over events $p_1, p_2, \ldots, p_n$ has the following properties:

- H is continuous in the p_i which allows small changes in probability to have small changes in entropy.
- If all the p_i are equal ($p_i = \frac{1}{n}$), then H monotonically increases with n, that is, the more equally likely events there are, the more uncertainty exists.
- If a choice can be broken down into two successive choices, the original H should be the weighted sum of the individual values of H. This requirement covers a special case which is not needed for our work.

The entropy of a distribution is given by Equation (2.5) which satisfies these properties [19]. From this equation we can determine that given two probability distributions with equal numbers of observations, the distribution which is flattest will have higher entropy.

$$H(P) = -\sum_{i=1}^{n} p_i \mathrm{log}_2 p_i \tag{2.5}$$

2.5.1 Principle of Maximum Entropy

The principle of maximum entropy (MaxEnt) was first introduced into modern literature by E. T. Jaynes but is essentially a formalization of Laplace's "Principle of Insufficient Reason". It states that the probability distribution where entropy is maximized subject to any known constraints provides the most unbiased representation of our knowledge of the system [20] [21]. Or, as Jaynes describes [22],

> ...the fact that a certain probability distribution maximizes entropy subject to certain constraints representing our incomplete information, is the fundamental property which justifies use of that distribution for inference; it agrees with everything that is known, but carefully avoids assuming anything that is not known. It is a transcription into mathematics of an ancient principle of wisdom...

We know from Equation (2.5), that maximizing the entropy in a system means finding the flattest probability distribution, which occurs when all probabilities have the same value. We provide a combinatorial proof that Equation (2.5) is maximized with a uniform distribution in Appendix A. While the maximum entropy possible in a system occurs when the probability distribution is flat, a flat distribution is not what we desire for a prediction system. We desire the flattest possible probability distribution subject to the constraints of the data.

2.5.2 Maximum Entropy Classifiers

The intuition behind MaxEnt classifiers is that models are initialized with a uniform distribution and are updated as more information becomes available through training data. If certain events are seen to be more likely in the training data, then they are weighted and the remaining probability mass is equally distributed across the rest of the distribution. For example, suppose we have a distribution with four possible outcomes. If we know nothing about the events then a MaxEnt classifier would assign a probability of $\frac{1}{4}$ to each event. If we know one of the events occurs $\frac{1}{2}$ of the time but we know nothing about the other three, a MaxEnt classifier would assign a probability of $\frac{1}{2}$ to the first event and $\frac{1}{6}$ to each of the other three. MaxEnt classifiers belong to a family of classifiers known as log-linear classifiers, which means they extract some set of features from the input and combine them linearly [14]. An argument for the reasonableness of using a MaxEnt classifier in the NLP domain is presented by Nigam *et al.* [23] where they showed that MaxEnt classifiers perform better than naive Bayes for some corpora. MaxEnt classifiers can be described in general using Equation (2.6)

$$p(c|d) = \frac{1}{Z}\text{exp}\left(\sum_i w_i f_i(c,d)\right) \tag{2.6}$$

where c is a class (a particular author in our case) and d is a given document. Z is a normalizing factor of the form

$$Z = \sum_C p(c|d) = \sum_{c' \in C} \text{exp}\left(\sum_{i=0}^{N} w_{c'i} f_i\right)$$

which forces the weights to sum to 1, $f_i(c,d)$ is an indicator function learned from the training data and w_i is a weighting. The probability distribution with maximum entropy represented by Equation (2.6) has been shown to be a unique distribution [24].

2.5.3 MEGAM

Once the corpus of documents has been converted into a set of feature vectors, the next step is to choose an implementation of MaxEnt to use for classification. For our work we chose a classifier developed by Duamé [25] called MEGA Model Optimization Package (MEGAM).

2.6 Dimensionality Manipulation

Our corpus consists of 18,862 documents written by 15 authors and containing 167,783 types (or unique words). Working with such a large data set proved to be difficult given the machine's memory limitations used for the computations. We used some well known concepts from linear algebra in order to reduce the dimensionality of the data set, and because we are treating each document as a vector and a collection of documents as a matrix, we were able to do PCA to reduce the dimensionality of our problem space. We used singular value decomposition (SVD) to perform PCA . While PCA and SVD are well known, implementing them on a computer was somewhat complex due to the large dimensions of our matrices.

2.6.1 Vector Projection

Vector projection is well known in linear algebra [26] but it is not commonly used in NLP. In general, given two vectors $\vec{A}$ and $\vec{B}$ projecting $\vec{A}$ onto $\vec{B}$ results in some vector $\vec{P}$ which is in the same direction as $\vec{B}$ but has length $||\vec{A}|| \cos\theta$ where θ is the angle between $\vec{A}$ and $\vec{B}$, and $||\vec{A}||$ is the 2-norm of $\vec{A}$. The 2-norm of a vector $\vec{A}$ is defined as $\sqrt{\langle\vec{A},\vec{A}\rangle}$.

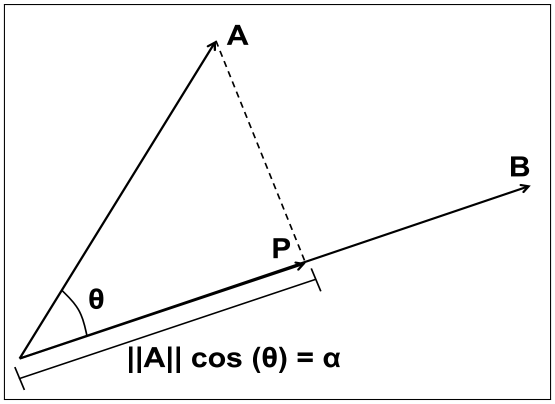

Figure 2.3: Vector Projection

Algebraically, the scalar projection of $\vec{A}$ onto $\vec{B}$ is given by

$$\alpha = \frac{\langle\vec{A},\vec{B}\rangle}{||\vec{B}||}$$

where $\langle\vec{A},\vec{B}\rangle$ denotes the inner product of $\vec{A}$ and $\vec{B}$. The vector projection of $\vec{A}$ onto $\vec{B}$ is given by

$$\vec{P} = \alpha \left(\frac{1}{||\vec{B}||} \cdot \vec{B} \right) = \frac{\langle \vec{A}, \vec{B} \rangle}{\langle \vec{B}, \vec{B} \rangle} \vec{B}. \tag{2.7}$$

Notice that since

$$\langle \vec{P}, \vec{P} \rangle = \left\langle \frac{\alpha}{||\vec{B}||} \cdot \vec{B}, \frac{\alpha}{||\vec{B}||} \cdot \vec{B} \right\rangle = \left(\frac{\alpha}{||\vec{B}||} \right)^2 \langle \vec{B}, \vec{B} \rangle = \frac{\alpha^2 \langle \vec{B}, \vec{B} \rangle}{\sqrt{\langle \vec{B}, \vec{B} \rangle}^2} = \alpha^2$$

and

$$\langle \vec{A}, \vec{P} \rangle = \frac{(\langle \vec{A}, \vec{B} \rangle)^2}{\langle \vec{B}, \vec{B} \rangle} = \alpha^2$$

it follows that

$$\langle \vec{A} - \vec{P}, \vec{P} \rangle = \langle \vec{A}, \vec{P} \rangle - \langle \vec{P}, \vec{P} \rangle = \alpha^2 - \alpha^2 = 0$$

which shows that $\vec{A} - \vec{P}$ is orthogonal to $\vec{P}$ and consequently to $\vec{B}$. If we suppose the document is represented by $\vec{A}$ and the topic of the document is represented by $\vec{B}$, then $\vec{A} - \vec{P}$ is completely independent from the topic. As shown above, $\vec{P}$ is the solution to Equation (2.1). It is our hypothesis that $\vec{A} - \vec{P}$ represents an author's style.

2.6.2 Singular Value Decomposition

Singular value decomposition [26] states that, given a $d \times v$ matrix M, there exists a factorization of the form

$$M = U_{d \times r} \Sigma_{r \times r} (V_{v \times r})^T$$

where U consists of the eigenvectors of MM^T, V consists of the eigenvectors of $M^T M$, and r is the rank of M. The columns of U are called left singular vectors and the columns of V are called right singular vectors. The matrix Σ is a diagonal matrix whose entries are singular values, or the square roots of the eigenvalues corresponding to the eigenvectors of U and V such that the diagonal entry $\sigma_{m,m}$, $1 \leq m \leq r$, is the singular value for the m^{th} columns of U and V.

In our setting, M is a *document* $\times$ *term* matrix and the matrices U and V are representations of M using orthonormal factors. Deerwester *et al.* [27] refer to these factors as "concepts" when describing latent semantic indexing (LSI). These orthonormal factors can be thought of as concepts in the following way. Assume all documents pertaining to a concept are linear combinations of each other, that is, they are linearly dependent, and then the number of concepts contained in a collection of documents is the maximal number of linearly independent rows (or columns) in the document matrix, which is known as the rank of the matrix. The rank of U represents the number of concepts present in the set of input documents, and each entry of U and will correspond to a document and a concept. The value in the entry can be thought of as a relative strength of the concept in the document. This thought process works similarly with concepts and vocabulary words using V. Here, U and V are often referred to as the document-to-concept similarity matrix and the term-to-concept similarity matrix respectively. We will develop a slightly deeper understanding of why these factors are referred to as concepts as we describe PCA.

2.6.3 Principal Component Analysis

Principal Component Analysis is an orthogonal linear transformation of a set of input vectors from the original d-dimensional space to a new k-dimensional space (ideally $k << d$) which results in a minimal loss of information [28]. The intuition behind this transformation and compression is that in a high dimensional vector space, there are dimensions in which the input data varies little. These dimensions therefore provide little to no information. If we could perform a change of basis on the vector space and remove the little-varying dimensions from the space, we would retain almost all of the information present in the original space but require lower dimensionality to represent it. Under PCA, basis vectors of the new space are ordered according to their relative importance based on how much information (variance along the dimension) that dimension provides. The first vector, called the first principal component, of the new space is the dimension along which the largest amount of variance from the input data lies. The first principal component is the dimension which, when the vectors in our space are projected onto it, has the greatest variance among the values. Figure 2.4 shows an example data set and the direction of its principal components.

The second principal component is the dimension that contains the second largest amount of variance. The remaining principal components are found similarly. We can think of each of these principal components as a vector representing the *concepts* in the input documents. Where a *concept* is a linear combination of the vocabulary words, the coefficients of which are given by some $\vec{t}$, as depicted in Figure 2.5. Since each of these *concepts* is a linear combination of words, the principal components are in the column space of A.

Figure 2.5 is a toy example of PCA. Given the vocabulary in our example, we can imagine that one of the PCA concepts could represent the idea of “vehicle”. The vehicle concept would be made up of some linear combination of the vocabulary words. In this example, by construction,

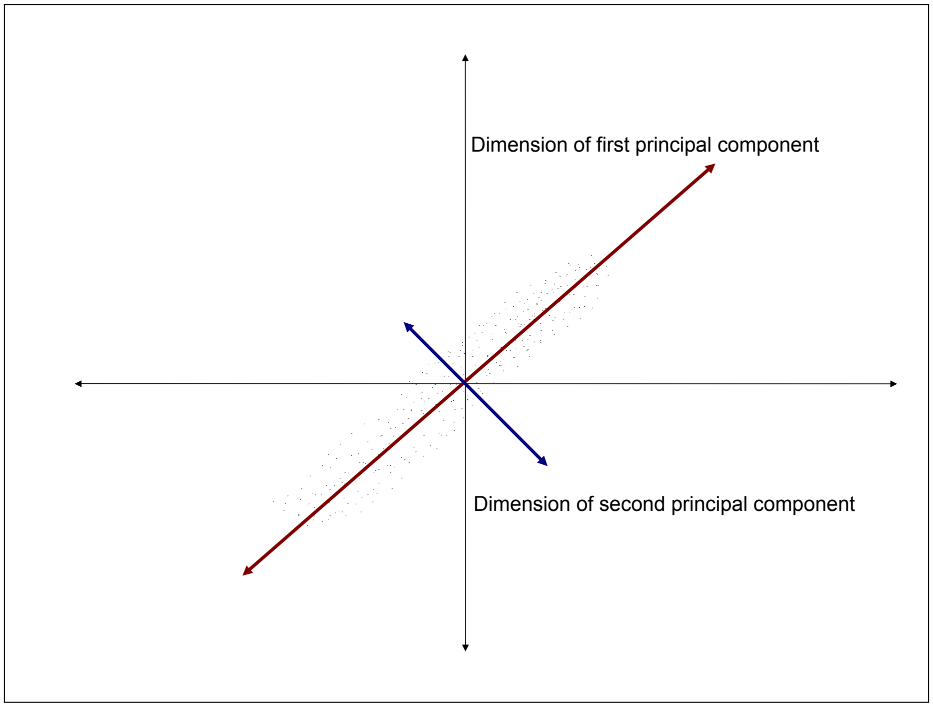
Dimension of first principal component
Dimension of second principal component

Figure 2.4: Dimensions of first and second principal components

$$\underbrace{\begin{array}{c|ccccc} & \text{Car} & \text{Book} & \text{Bus} & \text{Ghost} & \text{Truck} \\ \hline \text{Doc 1} & 5 & 7 & 3 & 19 & 0 \\ \text{Doc 2} & 4 & 2 & 14 & 8 & 6 \\ \text{Doc 3} & 15 & 18 & 23 & 7 & 1 \\ \text{Doc 4} & 6 & 9 & 0 & 1 & 6 \end{array}}_{A} \cdot \underbrace{\begin{array}{c} \text{Vehicle Concept} \\ \hline .3 \\ 0 \\ .4 \\ 0 \\ .9 \end{array}}_{t_i} = \begin{array}{c} \text{Strength of vehicle concept in each document} \\ \hline 2.7 \\ 12.2 \\ 14.6 \\ 7.2 \end{array}$$

Figure 2.5: Weights corresponding to the hypothetical concept of vehicles

we gave all of the weight to "vehicle" words. In an actual instance of PCA, words unrelated to the human concept "vehicle" may in fact get some weight. This might occur, for example, if the word "book" happened to co-occur often enough with "truck." After multiplying the input matrix by our vehicle concept vector, the result is a vector that tells us the relative strength of the vehicle concept present in each document. When actually performing PCA, the concepts are not given names but is done here for illustration purposes.

Consider the following example adapted from Leon [26] to see how PCA can be used to transform a space. Suppose we have d documents in a corpus with a total vocabulary of size v. Let A be a $d \times v$ matrix where each row corresponds to a document, each column corresponds to a word in our vocabulary, and the value in position $a_{i,j}$ represents the number of times word j occurs in document i. As a preparatory step, we calculate the mean value for each column of A. Let us call the mean value $\mu(v_j)$ which is, the average number of times word v_j occurs across all documents. Now, replace each element $a_{i,j}$ with $a_{i,j} - \mu(v_j)$. This preparatory step centers data in each column around the origin rather than the mean, which simplifies the variance calculation and the creation of the covariance matrix in the next step. It is often referred to as mean centering.

We wish to find mutually orthogonal vectors $\vec{y_1}$, $\vec{y_2}$, ... , $\vec{y_r}$ which correspond to the principal components of A to find a basis of our new space. Since principal components are in the column space of A, we can represent these vectors as a product where $\vec{y_i} = A\vec{t_i}$ for some $\vec{t_i} \in \mathbb{R}^n$. The first principal component, $\vec{y_1} = A\vec{t_1}$, is created by taking $\vec{t_1}$ as the solution to Equation (2.8). The second principal component will be created by solving Equation (2.8) subject to $\vec{t_i} \neq \vec{t_1}$,

$$\arg\max_{\vec{t_i}} \vec{t_i}^T S \vec{t_i} \tag{2.8}$$

where S is the covariance matrix of A, given by Equation (2.12) and the $\vec{t_i}$ are unit vectors. Recall the standard equations for the variance and covariance of random variables X, Y given by Equations (2.9) and (2.10). The denominator in these equations represents the degrees of freedom for the given sample. Often, these equations are given with $n - 1$ in the denominator rather than n. Variance calculated with $n - 1$ in the denominator is commonly called the *unbiased* variance.

$$var(X) = \frac{\sum\limits_{i=1}^{n} (x_i - \mu)^2}{n} \tag{2.9}$$

$$Cov(X,Y) = \sum_{i=1}^{n} \frac{(x_i - \mu_x)(y_i - \mu_y)}{n} \tag{2.10}$$

The degrees of freedom is simply the number of independent variables in a sample minus the number of estimated variables. Because we calculate the mean from the sample data (rather than knowing it *a-priori*), our degrees of freedom will be the number of samples minus one.

Since we subtracted the mean as a pre-processing step, $\mu = 0$ and the variance of $\vec{y_1}$ is given by Equation (2.11). The covariance matrix of A is given by Equation (2.12). In both equations, we divide by $v - 1$ rather than v where v is our vocabulary size. We divide by $v - 1$ because the mean is estimated from the data and so we have $v - 1$ degrees of freedom.

$$var(\vec{y_1}) = \frac{\langle \vec{y_1}, \vec{y_1} \rangle}{v - 1} = \frac{(A\vec{t_1})^T A\vec{t_1}}{v - 1} = \vec{t_1}^T S\vec{t_1} \tag{2.11}$$

$$S = \frac{A^T A}{v - 1}. \tag{2.12}$$

The solution to Equation (2.8) can be found by choosing $\vec{t_1}$ as the eigenvector of $A^T A$ corresponding to its maximum eigenvalue λ_1, which we now prove. What we want to show is that

$$\arg\max_{\vec{x}} \frac{\vec{x}^T A^T A\vec{x}}{\vec{x}^T \vec{x}} \tag{2.13}$$

is solved when $\vec{x}$ is the eigenvector which corresponds to the largest eigenvalue of $A^T A$. Equation (2.13) is known as the Rayleigh quotient. We begin with a simplifying step. By assuring that we scale all of the $\vec{x}$ vectors so that they are unit vectors prior to evaluating them, we force the denominator to be 1 and do not change the problem. Since $v - 1$ is just a scalar, we replace $A^T A$ with S. Now, our problem is to solve

$$\arg\max_{\vec{x}} \vec{x}^T S\vec{x}$$

subject to

$$\vec{x}^T \vec{x} = 1.$$

We use the method of Lagrange multipliers to solve this problem defining the function

$$f(\vec{x}, \lambda) = \vec{x}^T S\vec{x} - \lambda(\vec{x}^T \vec{x} - 1).$$

Taking the derivative of f with respect to $\vec{x}$ and λ we get the following equations.

$$\frac{df}{d\vec{x}} = \vec{x}^T S + \vec{x}^T S^T - 2\lambda\vec{x}^T = \vec{x}^T(S + S^T) - 2\lambda\vec{x}^T$$

and since S is symmetric

$$\frac{df}{d\vec{x}} = 2\vec{x}^T S - 2\lambda\vec{x}^T$$

and

$$\frac{df}{d\lambda} = -\vec{x}^T\vec{x} + 1.$$

Setting $\frac{df}{d\lambda} = 0$ we get our initial condition. Setting $\frac{df}{d\vec{x}} = 0$ and solving we get

$$2\vec{x}^T S = 2\lambda\vec{x}^T$$

$$\vec{x}^T S = \lambda\vec{x}^T$$

$$(\vec{x}^T S)^T = (\lambda\vec{x}^T)^T$$

$$S^T\vec{x} = \lambda\vec{x}$$

and since S is symmetric we get

$$S\vec{x} = \lambda\vec{x}. \tag{2.14}$$

Hence, the maximum of Equation (2.13) is found when $\vec{x}$ is an eigenvector corresponding to the eigenvalue λ. Now multiplying both sides of Equation (2.14) by $\vec{x}^T$ on the left we get

$$\vec{x}^T S\vec{x} = \vec{x}^T\lambda\vec{x}$$

and dividing both sides by the nonzero scalar $\vec{x}^T\vec{x}$ we have

$$\frac{\vec{x}^T S\vec{x}}{\vec{x}^T\vec{x}} = \lambda.$$

The left hand side of the equation above is maximized when $\vec{x}$ is the eigenvector corresponding to the largest eigenvalue of $A^T A$. Thus, $\vec{t_1}$ is the right singular vector of A corresponding to the largest singular value $\sigma_1 = \sqrt{\lambda_1}$. If $\vec{u_1}$ is the corresponding left singular vector (eigenvectors of AA^T), then

$$\vec{y_1} = A\vec{t_1} = \sigma_1\vec{u_1}$$

and similarly

$$\vec{y_2} = A\vec{t_2} = \sigma_2\vec{u_2}.$$

There are several methods for performing PCA [28] [29], and we choose to use SVD.

We accomplish dimensionality reduction by taking only the number of principal components required to account for 95% of the variance of our original matrix A. Since we are treating the eigenvalues as measures of variance, we can think of the sum of the eigenvalues as the total amount of variance in matrix A. We want to find the smallest j such that

$$\sum_{i=1}^{j} \sigma_i \geq .95 \sum_{i=1}^{v} \sigma_i.$$

If we take only the first j principal components of A and treat them as row vectors in a new $d \times j$ matrix A', then we have reduced the dimensionality while maintaining most of the "information" present in the original matrix A.

2.7 Evaluation Criteria

These are the metrics used to measure the results of our experiments.

2.7.1 Precision Recall and Accuracy

Two standard measures of success in the domain of NLP are *precision* and *recall*, where *precision* is a measure of true positive classifications in relation to the total number of positive classifications and *recall* is a measure of true positives in relation to the total number of actual positives. These metrics are primarily used when evaluating a binary classifier [14]. A third common measure is *accuracy*, which is mostly used in a multiclass problem and is a measure of the proportion of the number of correct classifications to the total sample size. We use the following equations

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

		Predicted	Value
		T	F
Actual	T	tp	fp
Value	F	fn	tn

Table 2.1: Confusion Matrix

where tp, tn, fp, fn are values from a confusion matrix, as depicted in Table 2.1.

For example, suppose our classification task was to determine whether or not prisoners are terrorists. Precision tells us the percentage of people we detained that are actually terrorists, and recall tells us the percentage of terrorists correctly identified.

2.7.2 F-Score

The last metric we use is the F-score, which is the harmonic mean of the precision and recall [14].

$$\text{F-score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$$

The reason for using F-score as a metric as opposed to accuracy is that it will not reward an increase in one dimension at the expense of the other. It is easy to manipulate your recall by sacrificing your precision. In our terrorist example, if we called everyone a terrorist, our recall would be 100% but our precision would drop. Measuring the F-score prevents these kinds of manipulations.

2.7.3 Weighted Metrics

When using novel topic cross-validation on an unbalanced data set, each division of data has different sizes since there are different numbers of documents in each topic category. As a result, we need to use a weighted average and standard deviation. The weights are computed as the fraction of the total document count represented within the cross-validation fold. A derivation for the unbiased variance estimate of a weighted average is provided by Schein *et al.* [11].

$$\hat{\mu} = \sum_{i=1}^{n} w_i x_i$$

$$V_2 = \sum_{i=1}^{n} w_i^2$$

$$\hat{\sigma}^2 = \frac{1}{1 - V_2} \sum_{i=1}^{n} w_i (x_i - \hat{\mu})^2$$

In the equations above, the x_i refer to an evaluation statistic such as an accuracy, precision, recall, or F-score. Also V_2 is simply a variable used in the calculation of the weighted variance.

2.8 Tools

2.8.1 Naval Postgraduate School Machine Learning Tools

The Naval Postgraduate School machine learning (NPSML) group has developed a suite of tools to facilitate machine learning in its NLP lab. This suite of tools is publicly available via the Internet [30]. It defines an NPSML format for data and serves as a pipeline to convert from the NPSML format to various third-party machine learning tools.

2.8.2 Maximum Entropy (GA) Model Optimization Package

The NPSML library provides a tool for converting files from the NPSML format to the MEGAM format which makes this MaxEnt classifier a natural choice. MEGAM is publicly available for download [25] and has no restrictions for academic use. The software takes a set of input vectors as an input for the model building phase and another set of input vectors for the evaluation/prediction phase. A maximum entropy distribution is created from the input vectors, and the distribution is applied to the test vectors to predict their class.

CHAPTER 3:
Experimental Setup

3.1 Source of Data

The data used in these experiments are a subset of the NYT Annotated Corpus, which is a collection of over 1.8 million XML documents representing nearly all NYT articles published between January 1987 and June of 2007. While the documents were hand tagged, not all metadata information fields were present in all documents. For example, only 48% of the documents contained tags for the author [31]. A subset of the corpus, which contained author and topic tags was selected for the experiment.

3.2 Data Selection and Preparation

We chose articles with a single author and written about a single topic in order to perform novel topic cross-validation. The subset consisted of 18,862 documents written by 15 authors on 23 topics. The subset was unbalanced across both authors and topics, meaning that authors did not write on all topics and authors did not write in equal amounts. Each author wrote on more than one topic and each author has at least 25 documents in each training set. We believe this data set is a reasonable representation of what may be encountered in the wild. All punctuation and capitalization was removed from the documents before testing. The work of selecting the subset and extracting the text from the XML was performed by Caver [32]. We are grateful for the work.

It is worth noting that the XML extraction method used by Caver [32] allowed the lead paragraph of some documents to be repeated. This was due to an inconsistency in the XML tagging in the corpus. No effort was made to correct or account for this inconsistency.

3.2.1 Methodology Overview

Using the same classifier for all of our methods gave common structure to each set of experiments regardless of the features used to represent the documents. We began by choosing a feature selection method and converting each document into a dense vector of feature count pairs. Once all documents were converted, we created the training and test splits. We used novel topic cross-validation for all of our experiments. As our data set had 23 topics, we created 23 train/test splits by choosing one topic at a time and all documents written about that topic became the test set. All documents on the remaining 22 topics became the training set. Once split, the training set was fed into MEGAM. We noticed during our experimentation that the MEGAM algorithm would occasionally converge too soon. To help avoid getting stuck in these local maxima, we forced MEGAM to repeat its iteration 100 times for each training set. We then ran the test set through MEGAM and recorded the results. We used the weighted average, standard deviation, and variance metrics described in Chapter 2 to evaluate each method. Two more-complicated approaches, PCA and projection, required more data manipulation before being fed into MEGAM but the same general approach was followed.

3.3 Feature Selection

We used several methods for representing the documents as vectors for classification. Unigrams, gappy bigrams, and orthogonal sparse bigrams with gaps ranging from zero to five were used for a total of 8 methods. We stored each document as a set of ⟨feature, count⟩ where the feature was the unigram, bigram, etc and the count was the number of times that feature occurred in the document. As the documents were read in, we assigned a unique integer to each new feature, starting with zero and incrementing. The document labeling allowed us to think of the documents as vectors with features as dimensions and the counts as the values for their corresponding dimension. We were able to save space by storing the documents as sparse vectors. A sparse vector is a vector that has had all of the zero count dimensions removed. If a feature does not occur in a document, we do not store the ⟨feature, count⟩ for that feature. Thus, all dimensions not explicitly enumerated in the vector are assumed to have values of zero. Since most documents contain only a fraction of the total vocabulary of the corpus, storing the documents as sparse vectors significantly lowers the storage requirements. When the dimensions with zero counts are explicitly enumerated, the vectors are called dense rather than sparse. Using sparse vectors as opposed to dense vectors can have a significant impact on the time required for MEGAM to converge.

Two more-complicated approaches of document representation were also used. We perform ed PCA on our unigram vectors, transforming them from vocabulary space to PCA concept space. We also performed vector projection operations on the unigram vectors as well as the PCA-transformed concept vectors. These methods are described in detail below.

3.3.1 PCA of Unigram Vectors

In order to perform PCA on the input documents, we began by transforming each document in the training set into a dense vector and created a matrix where each row is a dense document vector. Now we have a $d \times v$ matrix where d is the number of documents in the training set and v is the total vocabulary size for the training set. We call this matrix A. We then performed mean centering on A. Mean centering is required for PCA. It also removes most of the zeros from our matrix, and as a result, we can no longer take advantage of sparse matrix representations for storage. Our next step was to perform SVD on A and use the resulting matrices to perform dimensionality reduction and convert our documents from vocabulary space to PCA concept space. This conversion is done by creating a new matrix T consisting of the right singular vectors and the dimensions that correspond to 95% of the variance from A. The new matrix T can be thought of as a transformation matrix which converts document vectors from the vocabulary space into the PCA concept space. Figure 3.1 is a graphical representation of this process. To avoid training on test data, we performed PCA for each of the 23 train/test splits.

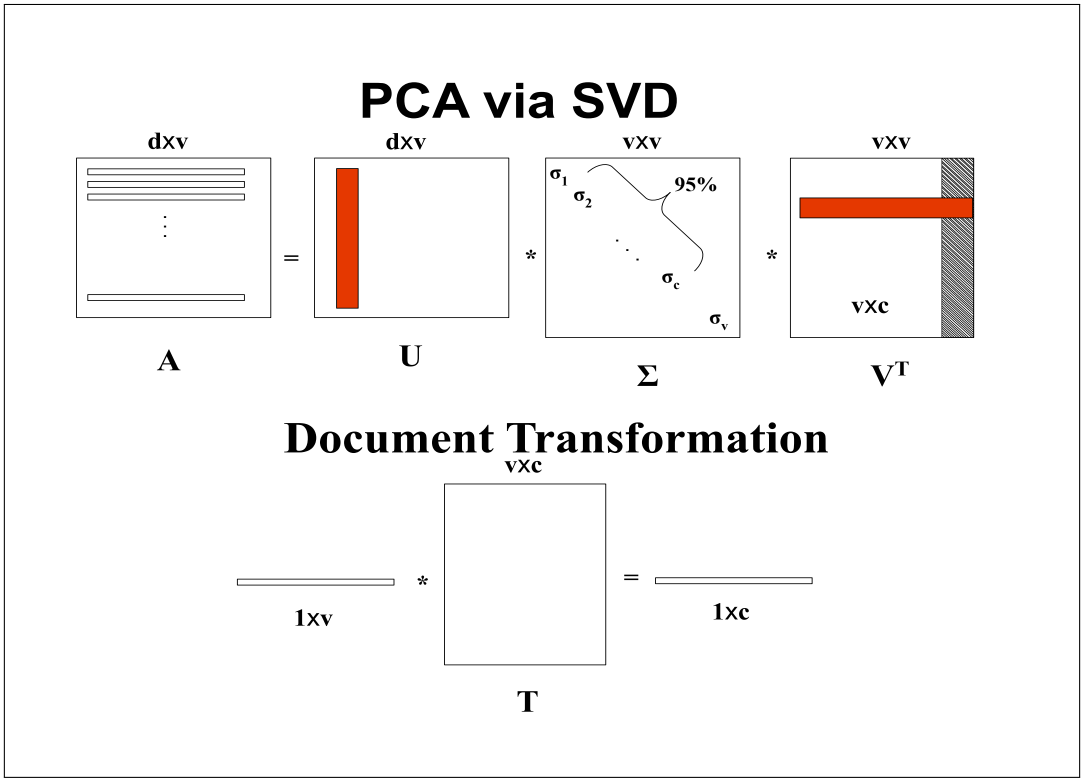

Figure 3.1: Red bars represent left and right singular vectors. Dashed region represents unused dimensions.

After creating the transformation matrix T, we multiplied each vector in the training and test sets by T to transform them into the PCA concept space. Of note, any words in the test set that were not present in the training set were ignored. The words were ignored because the dimensions of the dense vectors from the test and training sets needed to match for the matrix multiplication. The dimensions also needed to correspond to the same words for the transformation from vocabulary space to PCA concept space to have meaning. The exclusion of unseen words is justified because a MaxEnt classifier would not have any weight for an unseen word, and thus, an unseen word provides no predictive value for the document. It is these transformed test and training sets that we passed into MEGAM for training and classification.

PCA was successful as a dimensionality reduction tool. We could not store the PCA-transformed concept vectors as sparse vectors since the multiplication by the transform matrix put values in almost all of the dimensions; but, only keeping the dimensions of A that accounted for 95% of the variance resulted in significantly smaller vectors. This allowed for a reduction in input size from 18 GB to 1 GB and a significant decrease in the time required for classification as compared to the dense inputs from the projection of unigram vectors.

While SVD is a well known computation, we did have some difficulty implementing it due to the size of the matrices. For each of the training sets, the matrix A was approximately $18,000 \times 167,000$. Since we needed the right singular vectors for the dimensions of the transformation matrix to work, we had to compute A^TA which meant doing calculations with a $167,000 \times 167,000$ matrix or $27,889,000,000$ entries in the matrix. Since a 32-bit computer can only address $2^{32} = 4,294,967,296$ items, we had to do all the calculations using a 64-bit machine. The size of each A matrix was approximately 37 GB, which was too large of an input for the implementation we used. In order to reduce the A matrices sizes, for each of the $n - 1$ topics in the training set we took 20% of the documents at random and used this subset to construct reduced size A matrices. These smaller A matrices were used to create the T matrices. We multiplied all of the documents in the test set by T to create our transformed test set , not just the randomly selected 20% used to create T. We used the package SVDLIBC [33], to perform the calculations on our A matrices.

3.3.2 Projection of unigram and PCA-Transformed Concept Vectors

We were looking for a way to separate an author's style from the topic he is writing about. We want a vector to represent our document that is as close to the original document vector as possible, but has nothing in common with the topic vector. A solution to this problem is achieved by vector projection. The mechanics of projection are well known, and we used standard linear algebra formulas [26] for the computations.

We started by using unigram vectors of our documents, and we created a representative topic vector for each topic. The representative topic vector was created by taking the sum of all document vectors in that topic. We saved the topic vectors as dense vectors. We then expanded each sparse document into its dense representation, projected it onto its topic vector, and subtracted the result from the original document vector to get what we call the author vector. The process of projecting out the topic influence is represented in Figure 3.2. One difficulty we had with projecting the unigram vectors was storing and classifying the sets of author vectors after the projection operations. While the input document vector is a sparse vector, after the projection operation all of the dimensions have values, so we can no longer take advantage of sparse representations. This turned a 500 MB set of document vectors into an 18 GB set of author vectors. The increase in size also resulted in a significant increase in the time for MEGAM to process the training set. Each of the 23 iterations took about eight days to build a prediction model. We are fortunate to have access to a large machine with 48 processor cores and 256 GB of memory. Using this machine we were able to run several of these models in parallel.

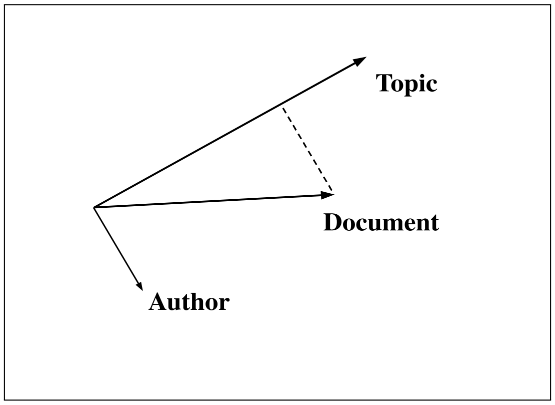

Figure 3.2: Projecting away the topic

As a means of overcoming the memory and computation difficulties that arose from the dense projected unigram vectors, we also applied vector projection to our PCA-transformed concept vectors. The process was very similar to projecting with unigram vectors except that all the projections had to be performed for each train/test split. We had to perform PCA for each train/test split because each transform matrix manipulated the input vectors in a different way, each training set had its own concept space. The topic vectors were created in the same manner, by taking the sum of all documents written on that topic. The dimensionality reduction of the PCA significantly reduced the size of our document vectors, from 18 GB in the case of unigram projections down to 1 GB in the case of PCA projections. The reduction in size also reduced the classification time from over eight days to less than 10 hours.

In total, we performed novel topic cross-validation on all of the document representation types listed below.

- **Unigrams** - traditional bag-of-word
- **Bigrams** - adjacent word pairs
- **Gappy Bigrams-1** - word pairs where the distance between words was at most one
- **Gappy Bigrams-2** - word pairs where the distance between words was at most two
- **Gappy Bigrams-3** - word pairs where the distance between words was at most three
- **Gappy Bigrams-4** - word pairs where the distance between words was at most four
- **Gappy Bigrams-5** - word pairs where the distance between words was at most five
- **OSB** -word pairs where the distance between words was at most five and the distance is added as a feature
- **PCA Unigram** - created by performing PCA on the unigram vectors
- **PCA Projection** - created by projecting the PCA unigram vectors
- **Unigram Projection** - created by projecting the unigram vectors

CHAPTER 4:
Results and Analysis

4.1 Overview

We used several methods for representing the documents and ran all of them through MEGAM. Novel topic cross-validation was used to test all of the representation methods. We computed the accuracy for each fold to come up with the statistics for each method, and then used the weighted average and weighted variation equations to compare the methods. Table 4.1 shows the weighted average accuracy and weighted standard deviation of all document representation methods. Table 4.2 shows the accuracy for each of the 23 topics for unigrams, PCA, projected unigrams, and projected PCA.

Table 4.1 gives the results for our experiments. One notable finding is the downward trend in accuracies for the first eight experiments. As the document representation method became more and more complex (moving from unigrams to OSBs), the author signal became more and more faint as compared to the topic signal and accuracy dropped. We believe the reduced accuracy is because the structure of the data set makes it inherently noisy and as the technique changed, the small amount of author signal originally present in each document was lost in the noise. We attempted two techniques to reduce the noise of the data set, PCA and projection.

4.1.1 Standard Methods

The standard methods for document representation we used were unigrams, bigrams, gappy bigrams, and orthogonal sparse bigrams with gaps ranging from one to five were used for a total of 8 methods. In order to show an improvement with our projection method, we needed to establish a baseline to compare against. Of the standard methods we tried, unigrams performed the best and we used them as our baseline method. The results of the other seven methods were poor, and we did not analyze them.

4.1.2 PCA of Unigram Vectors

We used PCA to transform the input documents from unigram vocabulary space to PCA concept space. The transformation kept the PCA concepts that accounted for 95% of the variance in the original documents. Our goal was to create a data set with less noise than the input data set while maintaining all of the important information. What we saw when we used the PCA

	Accuracy	Standard Deviation
Unigrams	0.7272	0.1631
Bigrams	0.5591	0.2798
Gappy Bigrams-1	0.4491	0.2806
Gappy Bigrams-2	0.2872	0.2146
Gappy Bigrams-3	0.4372	0.3125
Gappy Bigrams-4	0.3258	0.2853
Gappy Bigrams-5	0.3255	0.2834
OSB	0.3259	0.2841
PCA Unigram	0.7236	0.1719
PCA Projection	0.2951	0.3943
Unigram Projection	UNK	UNK

Table 4.1: Methods Applied and Results

concept documents to classify was almost exactly the same results as when using unigrams. This similarity in results is exactly what we should expect from a good PCA transformation. Our PCA dimensionality reduction process kept all of the *important* dimensions while discarding the dimensions that were simply noise. It is also worth noting that the accuracy was maintained even though we took a random 20% subset of each topic to compute our PCA transform matrix. This suggests that, for our data set, all documents on a given topic are sufficiently similar; all the PCA concepts present in the topic documents can be accurately determined by analyzing a small subset of the documents.

4.1.3 Projection of Unigram and PCA-Transformed Concept Vectors

We used vector projection in an attempt to tease apart the author signal and the topic signal in each document. We expected the accuracy of the classifications after performing the vector projection operations to have a significant impact. We did not expect the projections to make the results worse, but this is exactly what happened for both unigrams and PCA concept vectors.

Due to system issues, some of our unigram projection experiments did not finish. The topics that did not finish are listed in Table 4.2 as DNF. The weighted accuracy and standard deviation of the runs that did finish were 0.2206 and 0.1759 respectively. As a comparison, the weighted accuracy and standard deviation for unigrams were 0.6871 and 0.1142 respectively, when only considering the topics that the projected unigrams completed. These partial results indicate that the projection operations perform as poor on unigrams as they do on PCA-transformed concept vectors.

	Unigrams	**PCA**	**Proj. Uni.**	**Proj. PCA**
T50014	0.7282	0.7821	0.5468	0.0103
T50031	0.6717	0.7037	0.1850	0.1282
T50013	0.5559	0.5051	0.0168	0.0192
T50128	0.4587	0.4724	DNF	0.3911
T50012	0.7270	0.6494	0.1997	0.0057
T50048	0.7272	0.7289	0.3636	0.0028
T50015	0.9544	0.9596	0.0370	0.9961
T50097	0.8489	0.8800	DNF	0.6933
T50050	0.7064	0.7647	0.2926	0.0039
T50006	0.9429	0.8857	0.4517	0.3429
T50115	0.8668	0.8363	DNF	0.4027
T50136	0.8756	0.9365	DNF	0.9670
T50187	1.0000	1.0000	DNF	1.0000
T51556	0.8182	0.7636	0.0729	0.6000
T50172	0.9771	0.9798	DNF	1.0000
T50383	0.6747	0.5241	DNF	0.9398
T50368	0.8025	0.7840	DNF	0.0185
T50273	0.5699	0.5965	0.5384	0.0584
T50222	0.9917	1.0000	DNF	1.0000
T50338	0.8654	0.8526	0.1767	0.9872
T50049	1.0000	1.0000	0.0584	0.9844
T50214	0.9939	0.9939	DNF	0.9018
T50077	1.0000	1.0000	DNF	1.0000

Table 4.2: Accuracy by Topic (DNF indicates did not finish)

4.2 Analysis

One hypothesis as to why our projection method did so poorly, is inherent noise in the data set. We hypothesized we had noisy document vectors being projected onto noisy topic vectors and the resulting author vectors were therefore noise. A second hypothesis was that our topic vectors did not accurately represent the topic. If we project documents onto vectors that do not represent the topic, then we have no reason to believe we will end up with an author vector after the operation. To test these hypotheses, we ran the projection operations on our PCA concept vectors. If these projections had been successful then we could have concluded that noise was the cause of the poor performance. As shown in Tables 4.1 and 4.2, using projection on PCA concept vectors does worse than PCA and unigrams. We believe this rules out noise as the cause of the decreased performance.

One could also hypothesize that the author signal in a document is so strong that it overwhelms the topic signal and that by taking the union of all words used by all authors as the topic vector (the sum of all unigram document vectors on a topic) that you would end up with nothing but noise for a topic vector. If each document was mostly author signal then this is exactly what you would expect; however, we do not believe this is the case. Prior work suggests that topic signals can easily overwhelm author signals. For example, using the same data set Schein *et al.* [11] noted a statistically significant drop in accuracy when moving from a standard n-fold cross-validation to novel topic cross-validation. Their conclusion was that topic signals overwhelm author signals when performing authorship attribution. We believe our results support this conclusion. It still may be the case though, that taking the sum of all documents on a topic is not a suitable method for constructing a topic vector.

Since vector projection is well known, provably correct, and relatively easy to perform, we do not believe the problem lies with our implementation. Squared error is a standard distance metric. We proved that the solution to our problem of finding an author vector which is as close to the document vector as possible and still orthogonal to the topic vector is achieved through projection. Since the vectors are clearly not author vectors, at least one of our assumptions must be incorrect. We believe our results show that at least two of our assumptions were incorrect. Underlying assumptions we might question are:

1. Each author has a unique style which does not vary across topics
2. This style is completely distinct (orthogonal) from the topics he is writing about
3. Topic and author vectors are appropriately represented as feature-count vectors

It is commonly understood that different topics require different writing styles. People use different grammatical style, sentence structure, and vocabulary when writing a love note as opposed to writing a technical article. What we hoped to identify through the projections was the author's unique style, or fingerprint, what is often referred to as an author's voice [34]. If we can sufficiently capture an author's voice, then we should be able to create an author model that is somewhat invariant across topics. It is possible that an author's unique style is not constant across topics, as we had assumed in 1. Instead, it might be that the topic has some influence on the author's unique style. It also might be that the author consciously or unconsciously pushes his author vector towards the topic vector, that is, the author vector and topic vectors are not actually orthogonal, as we had assumed in 2. If an author pushes their author vector towards the topic vector, then their author vector is also not going to be unique across topics.

Another thing our results demonstrate is that n-grams do a good job of capturing topic signal, but do not do as well at capturing author signal. It makes sense that different topics require different vocabularies. One does not talk about extra points when writing about the ballet, but most likely would if writing about football. It should not be a surprise then that word counts do a much better job of capturing topic information than they do of capturing author information. What distinguishes an article I wrote from someone else's written on the same topic, then, is most likely not vocabulary. It might be that I use a more or less complex sentence structure, for example, and to capture the author signal one would need to capture long distance syntactic dependencies, as opposed to our third assumption.

4.2.1 Projection Anomalies

While overall, the projection operation performed worse than unigrams and PCA, there were several topics where the results were interesting. There were eight topics where the projections did worse than the maximum likelihood estimation (MLE) and ten topics where the projections did nearly as well as or better than both PCA and unigrams. The MLE was approximately 15% for all train/test splits.

Projection did better than PCA and unigrams in the following topics:

- Art (T50015)
- Restaurants (T50136)
- Advertising and Marketing (T50172)
- Golf (T50383)
- Soccer (T50338)

Projection accuracy was above 90% in the following topics:

- Appointments and Execuetive Changes (T50187)
- Photography (T50222)
- Suspensions, Dismissals and Resignations (T50049)
- Cooking and Cookbooks (T50214)

- Food (T50077)

Upon inspection of Table B.2, found in Appendix B, we noticed some similarities between the topics where projection did well which we believe explain the success. In all ten cases, all or the overwhelming majority of documents in the test set were written by one of four authors, A111915, A100046, A111487, and A111661. Interestingly, whenever these authors wrote on a topic, they were almost always the overwhelming majority author. Given our method for constructing a topic vector, when there is only one author writing on a topic then each document, as long as the author is consistent vis-á-vis that topic, is going to be very close to the topic vector. When an author vector is extracted from a document that is 'close to' its topic vector we end up with an author vector that is very close to the origin, as shown in Figure 4.1. What we have then, is four author models made up of vectors that are all very close to the origin. For all other authors, there were many other authors in their topics so their documents were not close to the topic vector and so their author vectors would be 'far' from the origin. Having vectors that were 'close' and 'far' from the origin created an artificial separation between these four authors and the other eleven. If an author vector was close to the origin it was one of the four, otherwise it was one of the eleven. It must be the case then, that these four authors were sufficiently different and their small author vectors were able to be differentiated between. We believe, our success in these cases is merely an artifact of the dataset.

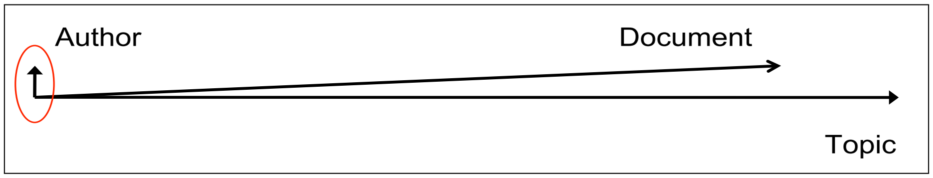

Figure 4.1: Example author vector when there is only one author writing on a topic

Projection did worse than MLE in the following topics:

- Books and Literature (T50014)
- Music (T50031)
- Baseball (T50013)
- Football (T50012)
- Motion Pictures (T50048)

- Dancing (T50050)
- Boxing (T50368)
- Horse Racing (T50273)

		Prediction						
		A100006	**A100023**	**A100078**	**A100024**	**A100068**	**A111487**	**A111915**
Truth	**A100006**	5	0	0	0	0	0	0
	A100023	741	0	11	0	0	0	0
	A100078	1599	0	3	0	0	0	0
	A100024	6	0	0	0	0	0	0
	A100068	524	0	15	0	0	0	0
	A111487	2	0	0	0	0	0	0
	A111915	0	0	1	0	0	0	0

Table 4.3: Confusion Matrix T50048

		Prediction				
		A100042	**A100046**	**A102480**	**A111554**	**A111915**
Truth	**A100042**	0	19	0	0	0
	A100046	0	780	0	0	0
	A102480	0	357	0	0	0
	A111554	0	775	0	0	0
	A111915	0	0	0	0	0

Table 4.4: Confusion Matrix T50115

What we see in Table 4.3 is an example of how easily the classifier is confused. A100006 was the author that MEGAM assigned to almost all of the candidate authors when evaluating the motion pictures topic (T50048), but he only wrote five articles on that topic. Interestingly, A100006 was the most prevalent author in the theater topic which could be considered close to motion pictures.

We see something similar in Table 4.4, except that the author who confused the classifier is one of the authors who is easily classified in other topics. Here, we are testing against Ice Hockey. All of the topics that A10046 wrote on were sports. It appears that MEGAM found similarities between topics and classified all candidate authors in the test set by the author with the strongest signal who also wrote on similar topics. Further analysis is necessary to see whether there are consistent reasons for this phenomenon.

4.3 Summary

We found that as our word-based document representation methods got more complex, the classification accuracy for our dataset degraded. We hypothesized that we could project out the topic influence on the document. We chose unigrams for our first attempt at projection since they had the highest accuracy among the different methods. The results of our projected unigrams were much worse than with unprojected unigrams. To test the hypothesis that the decrease in performance was due to noise in the dataset, we performed PCA on the input vectors to reduce the noise. We then tried to project out the topic from the PCA concept vectors. Overall, the PCA projections did poorly as well. While there were a few topics where PCA projections did better, we believe this was just an artifact of our dataset and not a validation of the method. We concluded that at least one of our assumptions must have been faulty. We hypothesize that an author's style vector is not orthogonal to the topic vector but is actually influenced by the topic vector. We believe authors, consciously or unconsciously, push their author vector towards the topic vector they are currently writing about. If it is the case that authors push their style vector towards the topic vector, then we cannot expect vector projection to work. By doing the projection what we are doing is creating noisy vectors and calling them an author vector. We should not be surprised at the poor performance.

CHAPTER 5:
Future Work and Conclusions

5.1 Summary

Our goal was to construct ⟨feature, count⟩ vectors of documents and topics in order to project out the influence of topic on each document and construct an author model that is relatively topic independent. We used a subset of the NYT Annotated Corpus where each document had a single author and was written on a single topic as our corpus. Our corpus had 18,862 documents written by 15 authors on 23 topics. We performed novel topic cross-validation on these documents, simulating the situation where we develop an author model and need to apply the model on a document written on a previously unseen topic. We used the following types of features to represent our documents: unigrams, bigrams, gappy-bigrams, OSBs, and PCA-transformed concept vectors. After converting documents into vectors we trained and classified the vectors using the MEGAM maximum entropy classifier. We compared our results to a standard 10 fold cross-validation and saw a statistically significant drop in accuracy from 98.35% to 72.72% [32][11]. The drop in accuracy shows how significant the impact a shift in topic domains has on the accuracy of current classification methods and provided us with a starting point. We also found, as the method of document representation became more complex, the weighted accuracy of the classifications degraded even further. Our results cast doubt on the validity of a widely held assumption in the field of NLP and further research and analysis is required to validate our conclusions.

5.2 Future Work

This research suggests a number of areas for future research which include the following:

- Perform a similar set of experiments using a different dataset.

  There is a possibility that the style the NYT editors demand affects the results. While it is difficult, if not impossible, to construct a dataset which does not have the influence of an editor, one could construct a dataset from multiple sources and thus minimize the impact of the editor's voice. An ideal dataset would be one where: each document was written by a single author on a single topic, there are multiple sources of the documents, each topic is represented in more than one document source, and each author is sufficiently represented in each training set. Constructing a dataset with these properties would not eliminate the impact of the editors, but it may minimize its impact on the results. This kind of dataset could also serve as an improved testbed for conducting authorship attribution research.

- Perform topic detection instead of author attribution

  We have hypothesized that feature-count vectors do a poor job of capturing authorship signal, but do a good job at capturing topic signal. This hypothesis could easily be tested by running a similar set of experiments and substituting author for topic. The experiments would then be novel author cross-validation. The dataset would need to be split up into train/test splits based on the author rather than the topic, and the documents would need to be labeled with their topic rather than their author prior to being classified. We would expect that the weighted accuracy for topic detection would be significantly higher than the weighted accuracy for author attribution. Good results when performing topic detection would validate our hypothesis that feature-count vectors do a much better job at capturing topic signal than they do at capturing author signal.

- Model authors using long-distance syntactic dependencies

  A significant area that we did not explore was conducting novel topic cross-validation where the author models are created using long-distance syntactic dependencies. One way to capture these dependencies would be regenerate our corpus from the source XML files, and not remove the capitalization or punctuation. With the full sentences, one could then feed the full sentences into a parser and use the parsed documents as data. Modeling the author using parsed documents may work better at capturing the author's unique style.

- Thoroughly analyze confusion matrices

  In several cases, MEGAM classified almost all authors in the test set as one author. In the two cases we reviewed, the author that MEGAM confused everyone with was the most prolific author in a topic similar to the tested topic, such as motion pictures and theater or ice hockey and other sports. More analysis would be necessary to determine if this was a consistent phenomenon or an aberration.

5.3 Conclusions

Our hypothesis was that we could project out the influence of topic on the documents, and thereby achieve similar accuracy using novel topic cross-validation as was seen when doing 10 fold cross-validation. If we would have been able to produce similar accuracy, we would have shown that we had a way of representing authors that could be applied across topic domains, and we would have been able to answer our question in the affirmative. What we saw instead, though, was our accuracy decreased significantly after attempting to project out the topic for both the unigrams and the PCA-transformed concept vectors. We concluded then, that we had made some faulty assumptions namely, that an author's style will not vary across topics, that an author's unique style vector will be orthogonal to the topic being written about, and that topic and author vectors can be appropriately represented as $\langle$feature, count$\rangle$ vectors.

The implications of these results are, while topics may be well represented by $\langle$feature, count$\rangle$ vectors, authors are not. Our results show that the topic drives the vocabulary being used, and it appears that the authorship signal we were searching for is not lexical. It may be that the authorship signal includes long-distance syntactic dependencies which we did not test. If the author signal is found in long-distance syntactic dependencies, then lexical models are better at modeling a topic than they are at modeling authors. By using lexical models, what we really are really doing then, is topic detection instead of authorship detection.

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

[1] F. Mosteller and D. Wallace, *Inference and Disputed Authorship: The Federalist*. Reading, MA: Addison-Wesley, 1964.

[2] G. Yule, "On sentence-length as a statistical characteristic of style in prose," *Biometrika Trust*, vol. 30, no. 1, pp. 363–390, 1939.

[3] K. Luyckx and W. Daelemans, "Authorship attribution and verification with many authors and limited data," *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 513–520, Jan 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1599146

[4] G. Gehrke, "Authorship discovery in blogs using bayesian classification with corrective scaling," Master's thesis, June 2008.

[5] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, March 2009. [Online]. Available: http://www3.interscience.wiley.com/journal/121572026/abstract

[6] G. Mikros and E. Argiri, "Investigating topic influence in authorship attribution," *Proceedings of the SIGIR '07 Workshop on Plagarism Analysis, Authorship Inentification, and Near-Duplicate Detection*, July 2007.

[7] M. Koppel, J. Schler, and E. Bonchek-Dokow, "Measuring differentiability: Unmasking pseudonymous authors," *Journal of Machine Learning Research*, vol. 8, no. 1, pp. 1261–1276, 2008.

[8] M. Corney, "Analysing e-mail text authorship for forensic purposes," Master's thesis, Queensland University of Technology, 2003.

[9] D. Madigan, A. Genkin, D. Lewis, S. Argamon, D. Fradkin, and L. Ye, "Author identification on the large scale," *Proceedings of the Meeting of the Classification Society of North America*, 2005.

[10] H. Baayen, H. Halteren, A. Neijt, and F. Tweedie, "Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution," *Literary and Linguistic Computing*, vol. 11, no. 3, pp. 121–132, 1996.

[11] A. Schein, J. Caver, R. Honaker, and C. Martell, "Author attribution evaluation with novel topic cross-validation," *Proceedings of the International Joint Conference on Knowledge Discovery...*, 2010.

[12] A. Schein, "Lagrange multipliers proof," 2010, personal correspondence between Randy Honaker and Andrew Schein.

[13] E. Stewart, *Calculus Early Transcendentals*, 3rd ed. 511 Forrest Lodge Road, Pacific Grove, CA 93950: Brooks/Cole, 1995.

[14] D. Jurafsky and J. Martin, *Speech and Language Processing*, 2nd ed. Pearson Education, 2009.

[15] D. Bikel and J. Sorensen, "If we want your opinion," 2007, pp. 493–500.

[16] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[17] C. Siefkes, F. Assis, and S. Chhabra, "Combining winnow and orthogonal sparse bigrams for incremental spam filtering," *Knowledge Discovery in Databases*, vol. 3202, pp. 410–421, Jan 2004. [Online]. Available: http://www.springerlink.com/index/89NTD9NYU544T58G.pdf

[18] G. Cormack, J. Gómez, and E. Sánz, "Spam filtering for short messages," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, ser. CIKM '07. New York, NY, USA: ACM, 2007, pp. 313–320. [Online]. Available: http://doi.acm.org/10.1145/1321440.1321486

[19] C. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, pp. 3–55, January 2001. [Online]. Available: http://doi.acm.org/10.1145/584091.584093

[20] E. Jaynes, "Information theory and statistical mechanics," *The Physical Review*, vol. 106, no. 4, pp. 620–630, May 1957.

[21] ——, “Information theory and statistical mechanics. ii,” *The Physical Review*, vol. 108, no. 2, pp. 171–190, October 1957.

[22] ——, “Notes on present status and future prospects,” *Maximum Entropy and Bayesian Methods*, pp. 1–13, 1991.

[23] K. Nigam, J. Lafferty, and A. McCallum, “Using maximum entropy for text classification,” *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61–67, 1999.

[24] D. Pietra, J. Lafferty, R. Technol, and S. Brooks, “Inducing features of random fields,” *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 19, no. 4, pp. 380–393, 1997.

[25] H. Daumé III. (2011, January) Mega model optimization package. [Online]. Available: http://www.cs.utah.edu/~hal/megam/

[26] S. Leon, *Linear Algebra with Applications*, 7th ed. Pearson Education, 2007.

[27] S. Deerwester, S. Dumais, G. Furnas, T. Laundauer, and R. Harshman, “Indexing by lantent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[28] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. MIT Press, 2010.

[29] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2001.

[30] A. Schein. (2011, January) The naval postgraduate school machine learning library. [Online]. Available: http://sourceforge.net/projects/npsml/

[31] E. Sandhaus, “The new york times annotated corpus overview.”

[32] J. Caver, “Novel topic impact on authorship attribution,” Master’s thesis, Naval Postgraduate School, December 2009.

[33] D. Rohde. Dough rohde’s svd c library. Online. [Online]. Available: http://tedlab.mit.edu/~dr/SVDLIBC/#formats

[34] T. Frank and D. Wall, *Finding your writer’s voice*. St. Martin’s Press, 1994.

[35] S. Roman, *Introduction to Coding and Information Theory*. Springer-Verlag New York, New York: Springer, 1997.

APPENDIX A: Maximum Entropy Proof

We wish to show that given some set $S = \{x_1, x_2, \ldots x_n\}$ with probability distribution P, where the probability of x_i is p_i, the entropy $H(P)$ is maximized when the probability distribution is uniform, that is, when all $p_i \in P$ are equal. To do this, we must first show that equations (A.1) and (A.2) hold for all $x > 0$ [35]. We use $\ln$ to denote the natural logarithm.

$$\ln x \leq x - 1 \tag{A.1}$$

$$\log_2 x \leq \frac{x-1}{\ln 2}. \tag{A.2}$$

Proof of (A.1):
Let $f(x) = \ln x - x + 1$, then $f^{'}(x) = \dfrac{1}{x} - 1$. Since $f^{'}(x)$ is equal to zero only when $x = 1$ and $f(1)$ exists, then $f(x)$ has exactly one critical point which occurs at $x = 1$. Since $f^{'}(x) > 0$ for all $x < 1$ and $f^{'}(x) < 0$ for all $x > 1$, the critical point is a maximum. This shows that for all x,

$$f(x) \leq f(1)$$

$$\ln x - x + 1 \leq 0$$

$$\ln x \leq x - 1$$

and equality holds exactly when $x = 1$ which is what we desired to show.

Proof of (A.2):
This follows directly from (A.1) and the fact that $\log_2 x = \dfrac{\ln x}{\ln 2}$.

Now we are ready to proceed. We wish to show that given some set $S = \{x_1, x_2, \ldots x_n\}$ with probability distribution P, where the probability of x_i is p_i

$$\arg\max_P H(P) = P^*, \tag{A.3}$$

where P^* is the uniform probability distribution. That is, that the entropy $H(P)$, is maximized when $p_i = p_j$ for all $1 \leq i, j \leq n$, where entropy is defined by Equation (2.5) as

$$H(P) = -\sum_{i=1}^{n} p_i \mathrm{log}_2 p_i = \sum_{i=1}^{n} p_i \mathrm{log}_2 \frac{1}{p_i}.$$

Proof of (A.3):
Let $P = \{p_1, p_2, \ldots, p_n\}$ be the probability distribution for S and let $R = \{r_1, r_2, \ldots, r_n\}$ be the uniform distribution. Note that the entropy of a system with a uniform distribution is

$$\sum_{i=1}^{n} r_i \mathrm{log}_2 \frac{1}{r_i} = \sum_{i=1}^{n} \frac{1}{n} \mathrm{log}_2 \frac{1}{\frac{1}{n}} = \mathrm{log}_2 n.$$

Equation (A.2) tells us that

$$\sum_{i=1}^{n} p_i \mathrm{log}_2 \frac{r_i}{p_i} \leq \frac{1}{\ln 2} \sum_{i=1}^{n} p_i \left(\frac{r_i}{p_i} - 1 \right)$$

and

$$\frac{1}{\ln 2} \sum_{i=1}^{n} p_i \left(\frac{r_i}{p_i} - 1 \right) = \frac{1}{\ln 2} \sum_{i=1}^{n} (r_i - p_i) = \frac{1}{\ln 2} \left(\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} p_i \right) = 0$$

which means

$$\sum_{i=1}^{n} p_i \mathrm{log}_2 \frac{r_i}{p_i} \leq 0. \tag{A.4}$$

Writing $\mathrm{log}_2(r_i/p_i)$ as $\mathrm{log}_2(1/p_i) - \lg(1/r_i)$, substituting into Equation (A.4) and rearranging we get

$$\sum_{i=1}^{n} p_i \lg \frac{1}{p_i} \leq \sum_{i=1}^{n} p_i \lg \frac{1}{r_i} = \sum_{i=1}^{n} p_i \lg n = \lg n.$$

Since the entropy of a system with a uniform probability distribution is $\lg n$, this tells us that of all probability distributions, none give higher entropy than the uniform distribution which is our desired result.

APPENDIX B:
Author and Topic Table

Table B.1 shows the natural language descriptions of the topics used on our experiments. Table B.2 shows the number of documents each author wrote on each topic.

T50014	Books and Literature	T50187	Appointments and Executive Changes
T50031	Music	T51556	Deaths (Obituaries)
T50013	Baseball	T50172	Advertising and Marketing
T50128	Theatre	T50383	Golf
T50012	Football	T50368	Boxing
T50048	Motion Pictures	T50273	Horse Racing
T50015	Art	T50222	Photography
T50097	Basketball	T50338	Soccer
T50050	Dancing	T50049	Suspensions, Dismissals and Resignations
T50006	Television	T50214	Cooking and Cookbooks
T50115	Hockey, Ice	T50077	Food
T50136	Restaurants		

Table B.1: Topic Categories

					AUTHORS					
		A100024	**A100078**	**A111554**	**A111915**	**A100046**	**A100042**	**A113159**	**A102480**	**…**
TOPICS	**T50014**	3	4	0	4	0	1	0	0	
	T50031	1	1149	0	0	0	0	0	0	
	T50013	0	0	491	0	12	55	1022	729	
	T50128	26	509	0	0	0	0	0	0	
	T50012	0	0	6	0	21	867	135	13	
	T50048	6	1602	0	1	0	0	0	0	
	T50015	0	1	0	0	0	0	0	0	
	T50097	0	0	179	0	25	10	3	6	
	T50050	1536	6	0	0	0	0	0	0	
	T50006	9	6	0	12	0	0	0	0	
	T50115	0	0	781	0	780	19	0	357	
	T50136	0	0	0	0	0	0	0	0	
	T50187	0	0	0	290	0	0	0	0	
	T51556	0	16	0	1	0	0	0	0	
	T50172	0	0	0	1487	0	0	0	0	
	T50383	0	0	4	0	157	5	0	0	
	T50368	0	0	6	0	0	155	0	1	
	T50273	0	0	25	0	33	17	0	0	
	T50222	0	0	0	0	0	0	0	0	
	T50338	0	0	1	0	154	0	0	1	
	T50049	1	0	0	63	0	0	0	0	
	T50214	0	0	0	0	0	0	0	0	
	T50077	0	0	0	0	0	0	0	0	
	TOTALS	**1582**	**3293**	**1493**	**1858**	**1182**	**1129**	**1160**	**1107**	

Table B.2: Topic/Author Data Tabulation

				AUTHORS				
TOPICS	A100512	A111487	A100023	A101068	A100006	A111661	A111723	TOTALS
T50014	0	3	1	354	18	1	1	**390**
T50031	0	0	0	0	1	0	783	**1934**
T50013	560	0	0	0	0	0	0	**2869**
T50128	0	0	145	1	842	0	1	**1524**
T50012	2	0	0	0	0	0	0	**1044**
T50048	0	2	752	539	5	0	0	**2907**
T50015	0	764	1	0	0	1	0	**767**
T50097	2	0	0	0	0	0	0	**225**
T50050	0	0	0	0	1	0	0	**1543**
T50006	0	0	3	0	2	1	2	**35**
T50115	0	0	0	0	0	0	0	**1937**
T50136	0	0	0	0	0	394	0	**394**
T50187	0	0	0	0	0	0	0	**290**
T51556	0	5	0	0	0	0	33	**55**
T50172	0	0	0	0	0	0	0	**1487**
T50383	0	0	0	0	0	0	0	**166**
T50368	0	0	0	0	0	0	0	**162**
T50273	490	0	0	0	0	0	0	**565**
T50222	0	121	0	0	0	0	0	**121**
T50338	0	0	0	0	0	0	0	**156**
T50049	0	0	0	0	0	0	0	**64**
T50214	0	0	0	0	0	163	0	**163**
T50077	0	0	0	0	0	64	0	**64**
TOTALS	**1054**	**895**	**902**	**894**	**869**	**624**	**820**	**18862**

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
Confusion Matrices For PCA Unigrams

Prediction

		A100006	**A100023**	**A100078**	**A100024**	**A111487**	**A111915**	**A111661**	**A111723**
Truth	**A100006**	2	0	0	0	0	0	0	0
	A100023	0	2	1	0	0	0	0	0
	A100078	0	0	5	0	1	0	0	0
	A100024	0	0	0	9	0	0	0	0
	A111487	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	12	0	0
	A111661	0	0	0	0	0	1	0	0
	A111723	0	0	1	0	0	0	0	1

Table C.1: Confusion Matrix T50006

Prediction

		A100006	**A111723**	**A100023**	**A111487**	**A100024**	**A100042**	**A100078**	**A101068**	**A111915**	**A111661**
Truth	**A100006**	16	2	0	0	0	0	0	0	0	0
	A111723	0	1	0	0	0	0	0	0	0	0
	A100023	0	0	0	1	0	0	0	0	0	0
	A111487	0	0	0	3	0	0	0	0	0	0
	A100024	0	0	0	0	3	0	0	0	0	0
	A100042	0	0	0	0	0	1	0	0	0	0
	A100078	0	0	0	0	0	0	4	0	0	0
	A101068	26	3	8	7	10	3	23	274	0	0
	A111915	0	0	0	0	0	0	1	1	2	0
	A111661	0	0	0	0	0	0	0	0	0	1

Table C.2: Confusion Matrix T50014

Prediction

		A100006	A100078	A100024	A111723	A100042	A101068	A102480	A100023	A111487	A111661	A100512	A113159	A111554	A111915
Truth	**A100006**	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	A100078	4	1108	23	2	1	3	1	2	1	4	0	0	0	0
	A100024	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	A111723	7	372	119	252	5	4	1	2	6	8	3	1	1	2
	A100042	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table C.3: Confusion Matrix T50031

Prediction

		A100006	A100023	A100078	A100024	A111487	A101068	A100042	A111661	A102480	A111915
Truth	**A100006**	5	0	0	0	0	0	0	0	0	0
	A100023	18	563	147	16	4	3	1	0	0	0
	A100078	36	62	1473	4	13	14	0	0	0	0
	A100024	0	0	1	4	0	0	1	0	0	0
	A111487	0	0	0	0	2	0	0	0	0	0
	A101068	46	228	164	19	5	71	2	3	1	0
	A100042	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	1

Table C.4: Confusion Matrix T50048

Prediction

		A100006	A100024	A100078	A100046	A111487	A101068	A100042	A100512	A111915	A100023	A111723	A102480	A111661
Truth	**A100006**	1	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	38	1173	173	2	14	14	3	1	5	7	100	3	3
	A100078	0	0	6	0	0	0	0	0	0	0	0	0	0
	A100046	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0

Table C.5: Confusion Matrix T50050

Prediction

		A100006	A100024	A100078	A100023	A111661	A102480	A101068	A111487	A100042	A111723	A111915
Truth	**A100006**	76	233	362	123	4	2	15	19	3	5	0
	A100024	0	25	1	0	0	0	0	0	0	0	0
	A100078	0	15	487	3	0	1	1	0	0	1	1
	A100023	0	6	6	131	1	1	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	1	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	1	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0

Table C.6: Confusion Matrix T50128

		Prediction						
		A100023	**A100078**	**A111487**	**A101068**	**A111661**	**A100024**	**A100006**
Truth	**A100023**	1	0	0	0	0	0	0
	A100078	0	1	0	0	0	0	0
	A111487	1	19	733	3	4	3	1
	A101068	0	0	0	0	0	0	0
	A111661	0	0	0	0	1	0	0
	A100024	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0

Table C.7: Confusion Matrix T50015

		Prediction	
		A100024	**A111915**
Truth	**A100024**	1	0
	A111915	0	63

Table C.8: Confusion Matrix T50049

		Prediction									
		A100042	**A100046**	**A102480**	**A111554**	**A100512**	**A111915**	**A100078**	**A113159**	**A111723**	**A100006**
Truth	**A100042**	613	159	40	47	6	1	1	0	0	0
	A100046	0	18	1	2	0	0	0	0	0	0
	A102480	0	1	11	1	0	0	0	0	0	0
	A111554	1	1	0	4	0	0	0	0	0	0
	A100512	0	0	0	0	2	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0
	A113159	7	42	11	32	8	0	0	30	3	2
	A111723	0	0	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0	0

Table C.9: Confusion Matrix T50012

Prediction

		A100042	A111554	A100046	A100078	A111915	A100512	A111723	A111661	A102480	A100006	A100024	A113159	A101068
Truth	**A100042**	46	3	4	1	1	0	0	0	0	0	0	0	0
	A111554	67	355	42	0	0	1	3	0	21	0	0	1	1
	A100046	1	0	11	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	94	11	73	0	0	348	12	1	21	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	38	43	72	0	2	3	2	1	561	3	3	1	0
	A100006	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	185	112	267	1	2	126	38	5	138	6	13	128	1
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0

Table C.10: Confusion Matrix T50013

Prediction

		A100042	A100046	A102480	A100512	A111554	A101068	A113159
Truth	**A100042**	9	1	0	0	0	0	0
	A100046	0	23	2	0	0	0	0
	A102480	0	0	6	0	0	0	0
	A100512	1	0	0	1	0	0	0
	A111554	4	4	14	0	156	1	0
	A101068	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	3

Table C.11: Confusion Matrix T50097

Prediction

		A100042	A100046	A102480	A100512	A111554	A113159	A100023	A101068	A111661	A100024	A111723	A100078
Truth	**A100042**	16	1	1	1	0	0	0	0	0	0	0	0
	A100046	22	730	8	2	17	1	0	0	0	0	0	0
	A102480	23	52	244	2	35	1	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	66	63	9	2	630	4	2	1	1	1	1	1
	A113159	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0	0	0

Table C.12: Confusion Matrix T50115

Prediction

		A100042	A102480	A111554	A101068	A100024	A100046	A111661	A100006	A100512	A111915	A100078	A111723	A100023	A113159
Truth	**A100042**	6	2	3	2	1	2	1	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	1	1	15	0	2	6	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100046	1	0	0	0	0	31	0	1	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	41	11	4	0	13	103	0	0	285	15	9	6	1	2
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table C.13: Confusion Matrix T50273

Prediction

		A100042	A100046	A113159	A111554	A102480	A101068	A100023	A100078
Truth	**A100042**	122	6	3	15	6	1	1	1
	A100046	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0
	A111554	0	0	0	4	1	1	0	0
	A102480	0	0	0	0	1	0	0	0
	A101068	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0

Table C.14: Confusion Matrix T50368

Prediction

		A100042	A100006	A100046	A111554	A100512	A102480	A100078
Truth	**A100042**	4	1	0	0	0	0	0
	A100006	0	0	0	0	0	0	0
	A100046	62	0	82	5	3	3	2
	A111554	3	0	0	1	0	0	0
	A100512	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0

Table C.15: Confusion Matrix T50383

Prediction

		A100046	A100042	A111554	A102480
Truth	**A100046**	131	21	2	0
	A100042	0	0	0	0
	A111554	0	0	1	0
	A102480	0	0	0	1

Table C.16: Confusion Matrix T50338

Prediction

		A111915	A100078	A111723	A111661	A111487	A101068	A100023	A100024	A102480	A100042
Truth	**A111915**	1457	13	2	4	2	4	1	2	1	1
	A100078	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0

Table C.17: Confusion Matrix T50172

Prediction

		A111915
Truth	**A111915**	290

Table C.18: Confusion Matrix T50187

Prediction

		A100078	A111723	A100024	A111487	A111915
Truth	**A100078**	14	1	1	0	0
	A111723	6	24	3	0	0
	A100024	0	0	0	0	0
	A111487	1	0	1	3	0
	A111915	0	0	0	0	1

Table C.19: Confusion Matrix T51556

		Prediction
		A111487
Truth	**A111487**	121

Table C.20: Confusion Matrix T50222

		Prediction
		A111661
Truth	**A111661**	64

Table C.21: Confusion Matrix T50077

		Prediction									
		A111661	**A100024**	**A100042**	**A100078**	**A100023**	**A111487**	**A100006**	**A100512**	**A102480**	**A111915**
Truth	**A111661**	369	5	8	4	3	1	1	1	1	1
	A100024	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0

Table C.22: Confusion Matrix T50136

		Prediction	
		A111661	**A111723**
Truth	**A111661**	162	1
	A111723	0	0

Table C.23: Confusion Matrix T50214

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D:
Confusion Matrices For PCA Projection

Prediction

		A100006	**A111915**	**A100023**	**A100024**	**A100078**	**A111661**	**A111723**
Truth	**A100006**	0	2	0	0	0	0	0
	A111915	0	12	0	0	0	0	0
	A100023	0	3	0	0	0	0	0
	A100024	0	9	0	0	0	0	0
	A100078	0	6	0	0	0	0	0
	A111661	0	1	0	0	0	0	0
	A111723	0	2	0	0	0	0	0

Table D.1: Confusion Matrix T50006

Prediction

		A100006	**A100078**	**A100023**	**A100024**	**A100042**	**A101068**	**A111915**	**A111487**	**A111661**	**A111723**
Truth	**A100006**	0	18	0	0	0	0	0	0	0	0
	A100078	0	4	0	0	0	0	0	0	0	0
	A100023	0	1	0	0	0	0	0	0	0	0
	A100024	0	3	0	0	0	0	0	0	0	0
	A100042	0	1	0	0	0	0	0	0	0	0
	A101068	0	354	0	0	0	0	0	0	0	0
	A111915	0	4	0	0	0	0	0	0	0	0
	A111487	0	3	0	0	0	0	0	0	0	0
	A111661	0	1	0	0	0	0	0	0	0	0
	A111723	0	1	0	0	0	0	0	0	0	0

Table D.2: Confusion Matrix T50014

Prediction

		A100006	A100024	A100078	A111723
Truth	**A100006**	0	1	0	0
	A100024	0	1	0	0
	A100078	0	902	247	0
	A111723	0	680	103	0

Table D.3: Confusion Matrix T50031

Prediction

		A100006	A100023	A100078	A100024	A101068	A111487	A111915
Truth	**A100006**	5	0	0	0	0	0	0
	A100023	741	0	11	0	0	0	0
	A100078	1599	0	3	0	0	0	0
	A100024	6	0	0	0	0	0	0
	A101068	524	0	15	0	0	0	0
	A111487	2	0	0	0	0	0	0
	A111915	0	0	1	0	0	0	0

Table D.4: Confusion Matrix T50048

Prediction

		A100006	A100078	A100024
Truth	**A100006**	0	1	0
	A100078	0	6	0
	A100024	0	1536	0

Table D.5: Confusion Matrix T50050

		Prediction					
		A100006	A100078	A100023	A100024	A101068	A111723
Truth	A100006	122	720	0	0	0	0
	A100078	35	474	0	0	0	0
	A100023	78	67	0	0	0	0
	A100024	7	19	0	0	0	0
	A101068	1	0	0	0	0	0
	A111723	0	1	0	0	0	0

Table D.6: Confusion Matrix T50128

		Prediction			
		A100023	A111487	A100078	A111661
Truth	A100023	0	1	0	0
	A111487	0	764	0	0
	A100078	0	1	0	0
	A111661	0	0	1	0

Table D.7: Confusion Matrix T50015

		Prediction	
		A100024	A111915
Truth	A100024	0	1
	A111915	0	63

Table D.8: Confusion Matrix T50049

		Prediction						
		A100042	A111554	A111915	A100046	A100512	A102480	A113159
Truth	**A100042**	0	866	1	0	0	0	0
	A111554	0	6	0	0	0	0	0
	A111915	0	0	0	0	0	0	0
	A100046	0	21	0	0	0	0	0
	A100512	0	2	0	0	0	0	0
	A102480	0	13	0	0	0	0	0
	A113159	0	135	0	0	0	0	0

Table D.9: Confusion Matrix T50012

		Prediction					
		A100042	A100046	A100512	A102480	A111554	A113159
Truth	**A100042**	55	0	0	0	0	0
	A100046	12	0	0	0	0	0
	A100512	560	0	0	0	0	0
	A102480	729	0	0	0	0	0
	A111554	491	0	0	0	0	0
	A113159	1022	0	0	0	0	0

Table D.10: Confusion Matrix T50013

		Prediction					
		A100042	A111554	A102480	A100046	A100512	A113159
Truth	**A100042**	0	9	1	0	0	0
	A111554	0	152	27	0	0	0
	A102480	0	2	4	0	0	0
	A100046	0	24	1	0	0	0
	A100512	0	2	0	0	0	0
	A113159	0	1	2	0	0	0

Table D.11: Confusion Matrix T50097

Prediction

		A100042	A100046	A102480	A111554	A111915
Truth	**A100042**	0	19	0	0	0
	A100046	0	780	0	0	0
	A102480	0	357	0	0	0
	A111554	0	775	0	0	6
	A111915	0	0	0	0	0

Table D.12: Confusion Matrix T50115

Prediction

		A100042	A100046	A111915	A100512	A111554
Truth	**A100042**	0	10	7	0	0
	A100046	0	33	0	0	0
	A111915	0	0	0	0	0
	A100512	0	490	0	0	0
	A111554	0	20	5	0	0

Table D.13: Confusion Matrix T50273

Prediction

		A100042	A113159	A102480	A111554
Truth	**A100042**	3	152	0	0
	A113159	0	0	0	0
	A102480	0	1	0	0
	A111554	0	6	0	0

Table D.14: Confusion Matrix T50368

		Prediction			
		A100042	**A100046**	**A111915**	**A111554**
Truth	**A100042**	0	5	0	0
	A100046	0	156	1	0
	A111915	0	0	0	0
	A111554	0	4	0	0

Table D.15: Confusion Matrix T50383

		Prediction		
		A100046	**A102480**	**A111554**
Truth	**A100046**	154	0	0
	A102480	1	0	0
	A111554	1	0	0

Table D.16: Confusion Matrix T50338

		Prediction
		A111915
Truth	**A111915**	1487

Table D.17: Confusion Matrix T50172

		Prediction
		A111915
Truth	**A111915**	290

Table D.18: Confusion Matrix T50187

Prediction

		A100078	A111723	A111487	A111915
Truth	**A100078**	0	16	0	0
	A111723	0	33	0	0
	A111487	0	5	0	0
	A111915	0	1	0	0

Table D.19: Confusion Matrix T51556

Prediction

		A111487
Truth	**A111487**	121

Table D.20: Confusion Matrix T50222

Prediction

		A111661
Truth	**A111661**	64

Table D.21: Confusion Matrix T50077

Prediction

		A111661	A111915
Truth	**A111661**	381	13
	A111915	0	0

Table D.22: Confusion Matrix T50136

Prediction

		A111661	A100078
Truth	**A111661**	147	16
	A100078	0	0

Table D.23: Confusion Matrix T50214

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: Confusion Matrices For Unigrams

		Prediction						
		A100006	A100023	A100078	A100024	A111915	A111661	A111723
Truth	**A100006**	2	0	0	0	0	0	0
	A100023	0	2	1	0	0	0	0
	A100078	0	0	6	0	0	0	0
	A100024	0	0	0	9	0	0	0
	A111915	0	0	0	0	12	0	0
	A111661	0	0	0	0	1	0	0
	A111723	0	0	0	0	0	0	2

Table E.1: Confusion Matrix T50006

		Prediction									
		A100006	A100042	A111723	A100023	A111487	A100024	A100078	A101068	A111915	A111661
Truth	**A100006**	16	1	1	0	0	0	0	0	0	0
	A100042	0	1	0	0	0	0	0	0	0	0
	A111723	0	0	1	0	0	0	0	0	0	0
	A100023	0	0	0	0	1	0	0	0	0	0
	A111487	0	0	0	0	3	0	0	0	0	0
	A100024	0	0	0	0	0	3	0	0	0	0
	A100078	0	0	0	0	0	0	4	0	0	0
	A101068	38	1	3	8	13	11	27	253	0	0
	A111915	0	0	0	0	0	0	1	1	2	0
	A111661	0	0	0	0	0	0	0	0	0	1

Table E.2: Confusion Matrix T50014

		Prediction													
		A100006	A100078	A100024	A111661	A100042	A102480	A100023	A111487	A111723	A101068	A100512	A113159	A111554	A111915
Truth	**A100006**	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	A100078	4	1110	23	5	1	1	1	1	1	2	0	0	0	0
	A100024	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	6	358	181	18	3	1	2	7	190	7	5	1	2	2
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table E.3: Confusion Matrix T50031

		Prediction										
		A100006	A111723	A100023	A100078	A111487	A100024	A101068	A100042	A111554	A111661	A111915
Truth	**A100006**	4	1	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0
	A100023	34	0	533	155	8	20	1	1	0	0	0
	A100078	35	1	40	1488	22	6	8	1	1	0	0
	A111487	0	0	0	0	2	0	0	0	0	0	0
	A100024	1	0	0	0	0	4	0	1	0	0	0
	A101068	59	1	157	201	10	24	81	1	0	5	0
	A100042	0	0	0	0	0	0	0	0	0	0	0
	A111554	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	1

Table E.4: Confusion Matrix T50048

		Prediction												
		A100006	**A100024**	**A100078**	**A101068**	**A111661**	**A100046**	**A111487**	**A111723**	**A100042**	**A111554**	**A111915**	**A100023**	**A102480**
Truth	**A100006**	1	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	45	1082	251	16	6	2	24	88	4	1	7	8	2
	A100078	0	0	6	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100046	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0

Table E.5: Confusion Matrix T50050

		Prediction										
		A100006	**A100024**	**A100078**	**A100023**	**A111661**	**A102480**	**A101068**	**A111487**	**A100042**	**A111723**	**A111915**
Truth	**A100006**	57	236	380	131	4	2	11	15	2	4	0
	A100024	1	24	1	0	0	0	0	0	0	0	0
	A100078	0	15	487	3	0	1	1	0	0	1	1
	A100023	0	6	7	130	1	1	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	1	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	1	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0

Table E.6: Confusion Matrix T50128

Prediction

		A100023	A100078	A111487	A111661	A100024	A101068	A111723	A100006
Truth	**A100023**	1	0	0	0	0	0	0	0
	A100078	0	1	0	0	0	0	0	0
	A111487	0	24	729	4	4	1	1	1
	A111661	0	0	0	1	0	0	0	0
	A100024	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0

Table E.7: Confusion Matrix T50015

Prediction

		A100024	A111915
Truth	**A100024**	1	0
	A111915	0	63

Table E.8: Confusion Matrix T50049

Prediction

		A100042	A102480	A111554	A100046	A100512	A111661	A111915	A100078	A113159	A100006	A111723	A100024
Truth	**A100042**	679	39	34	106	6	1	1	1	0	0	0	0
	A102480	0	13	0	0	0	0	0	0	0	0	0	0
	A111554	1	0	5	0	0	0	0	0	0	0	0	0
	A100046	0	2	2	17	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	2	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	6	14	26	34	8	0	0	0	44	1	1	1
	A100006	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0

Table E.9: Confusion Matrix T50012

Prediction

		A100042	A111554	A100512	A100046	A100078	A111915	A111723	A111661	A102480	A101068	A100024	A113159	A100006
Truth	**A100042**	46	2	1	4	1	1	0	0	0	0	0	0	0
	A111554	68	367	1	38	1	0	0	0	15	1	0	0	0
	A100512	63	9	387	75	0	0	6	1	19	0	0	0	0
	A100046	1	0	0	11	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0
	A102480	42	37	1	74	1	1	2	1	562	1	4	1	2
	A101068	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	226	83	93	249	3	2	26	7	107	1	10	211	4
	A100006	0	0	0	0	0	0	0	0	0	0	0	0	0

Table E.10: Confusion Matrix T50013

Prediction

		A100042	A100046	A102480	A100512	A111554	A113159	A101068
Truth	**A100042**	9	1	0	0	0	0	0
	A100046	0	23	2	0	0	0	0
	A102480	0	0	6	0	0	0	0
	A100512	1	0	0	1	0	0	0
	A111554	4	3	21	0	149	1	1
	A113159	0	0	0	0	0	3	0
	A101068	0	0	0	0	0	0	0

Table E.11: Confusion Matrix T50097

Prediction

		A100042	A100046	A100512	A111554	A102480	A113159	A111661
Truth	**A100042**	17	1	1	0	0	0	0
	A100046	23	724	3	21	9	0	0
	A100512	0	0	0	0	0	0	0
	A111554	40	43	1	682	9	3	3
	A102480	16	52	2	33	253	1	0
	A113159	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0

Table E.12: Confusion Matrix T50115

Prediction

		A100042	A102480	A111554	A100078	A100024	A100046	A111661	A100006	A100512	A111915	A111723	A100023	A113159	A111487
Truth	**A100042**	9	1	2	1	1	2	1	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111554	1	3	15	0	2	3	0	0	0	0	0	0	0	1
	A100078	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100046	1	0	0	0	0	31	0	1	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100512	41	14	4	11	18	103	2	0	269	19	6	1	2	0
	A111915	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table E.13: Confusion Matrix T50273

		Prediction							
		A100042	**A100046**	**A113159**	**A100512**	**A111554**	**A102480**	**A100023**	**A100078**
Truth	**A100042**	123	6	2	2	13	7	1	1
	A100046	0	0	0	0	0	0	0	0
	A113159	0	0	0	0	0	0	0	0
	A100512	0	0	0	0	0	0	0	0
	A111554	0	0	0	0	5	1	0	0
	A102480	0	0	0	0	0	1	0	0
	A100023	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0

Table E.14: Confusion Matrix T50368

		Prediction								
		A100042	**A101068**	**A100046**	**A100512**	**A111554**	**A100006**	**A102480**	**A100078**	**A111487**
Truth	**A100042**	4	1	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0
	A100046	35	0	104	4	7	1	2	3	1
	A100512	0	0	0	0	0	0	0	0	0
	A111554	1	0	0	0	3	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0

Table E.15: Confusion Matrix T50383

		Prediction				
		A100046	**A111915**	**A100042**	**A102480**	**A111554**
Truth	**A100046**	133	1	20	0	0
	A111915	0	0	0	0	0
	A100042	0	0	0	0	0
	A102480	0	0	0	1	0
	A111554	0	0	0	0	1

Table E.16: Confusion Matrix T50338

		Prediction									
		A111915	**A100078**	**A111723**	**A111661**	**A100024**	**A111487**	**A101068**	**A100023**	**A100042**	**A102480**
Truth	**A111915**	1453	19	1	3	3	2	2	1	2	1
	A100078	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0
	A111661	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0
	A101068	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0
	A102480	0	0	0	0	0	0	0	0	0	0

Table E.17: Confusion Matrix T50172

		Prediction
		A111915
Truth	**A111915**	290

Table E.18: Confusion Matrix T50187

		Prediction				
		A100078	**A111723**	**A111487**	**A100024**	**A111915**
Truth	**A100078**	15	1	0	0	0
	A111723	5	26	0	2	0
	A111487	1	0	3	1	0
	A100024	0	0	0	0	0
	A111915	0	0	0	0	1

Table E.19: Confusion Matrix T51556

Prediction

		A111487	A100024
Truth	**A111487**	120	1
	A100024	0	0

Table E.20: Confusion Matrix T50222

Prediction

		A111661
Truth	**A111661**	64

Table E.21: Confusion Matrix T50077

Prediction

		A111661	A100512	A100024	A100042	A100078	A100023	A111487	A100006	A111723	A111915
Truth	**A111661**	345	1	9	9	17	7	1	1	1	3
	A100512	0	0	0	0	0	0	0	0	0	0
	A100024	0	0	0	0	0	0	0	0	0	0
	A100042	0	0	0	0	0	0	0	0	0	0
	A100078	0	0	0	0	0	0	0	0	0	0
	A100023	0	0	0	0	0	0	0	0	0	0
	A111487	0	0	0	0	0	0	0	0	0	0
	A100006	0	0	0	0	0	0	0	0	0	0
	A111723	0	0	0	0	0	0	0	0	0	0
	A111915	0	0	0	0	0	0	0	0	0	0

Table E.22: Confusion Matrix T50136

Prediction

		A111661	A111723
Truth	**A111661**	162	1
	A111723	0	0

Table E.23: Confusion Matrix T50214

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. USCYBERCOM
   Fort George G Meade, Maryland

4. COMFLTCYBERCOM
   Fort George G Meade, Maryland